Quick Start Manual

Version 7

© 2006 PARIS Technologies, Inc.

Published March 2006

DISCLAIMER

Information in this document is subject to change without notice. Companies, names, and data used in examples are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of PARIS Technologies, Inc. The software described in this document is furnished under a license agreement and may be used only in accordance with the terms of such license.

All Products described or otherwise referenced herein are the property of their respective companies.

Published March 2006

Table of contents

Quick Start *1*

Quick Start Introduction **1**

Creating a PowerOLAP Database **2**

Creating Dimensions **6**

Adding Members to Dimensions **8**

Creating a Dimension Hierarchy **10**

The Dimension Hierarchy Toolbar 13

Aggregate Weights 13

Creating a Cube **14**

Creating a Slice View **16**

Arranging Slice Dimensions 19

Selecting Page Members 20

Changing the Grid Layout 22

Entering Data in a Slice **23**

Setting General & Format Preferences **23**

General Tab 23

Format Tab 27

Licensing Tab 29

Portal Tab 29

Importing Data from a File **31**

Creating Cube Formulas **34**

Creating an Excel Worksheet **39**

Selecting a Page Member to View in Excel 40

Entering Data from within Excel **41**

Defining Cube References **43**

Closing a Database **47**

Summary of Quick Start Exercises **48**

About OLAP Exchange **49**

About PowerOLAP Services **56**

PLEASE NOTE

Some exercises in this Quick Start manual and in the PowerOLAP User manual can not be performed with the Lite version of PowerOLAP client (Modeler) software.

Please contact PARIS Technologies to receive a temporary license for a full user version, which will allow you to follow all exercises in these manuals:

PARIS Technologies, Inc.
200 Hyde Park
Doylestown PA, 18901
(215) 340-2890
www.paristech.com

Quick Start

Note: Some of the Features such as Analysis Services, OLAP Exchange, and HTTP web connectivity are activated for use only when appropriate License is used. After activating the software, check Edit/Options/Licensing to view the functionality on your system.

Quick Start Introduction

The *PowerOLAP Quick Start Manual* is designed to introduce you to the fundamentals of PowerOLAP by using step-by-step procedures to create a PowerOLAP model. We recommend that new OLAP users become familiar with the *Introduction to PowerOLAP* as a prerequisite to this manual. The *Introduction* also contains information about Version 7 of PowerOLAP, including new features and their benefits.

By working through this *Quick Start Manual* you will learn many of PowerOLAP's elemental functions: how to create databases, Dimensions, Cubes and Slices; how to import data from a text file into a Cube; and how to create an Excel worksheet from a Slice view. This manual shows how to construct a model by working entirely with the product's "modeler." Note that—especially if you are an Administrator or power user—you will not build Cubes (business models, if you will) only by the steps described here. Instead, many PowerOLAP Cube features (Meta Data) will be built using the OLAP Exchange component; and, the actual figures you work with (Fact Data) will likely come from another source. However, you will undoubtedly need to understand the concepts and follow the steps covered in these pages because they convey how to use, customize and advance your own models. These skills, in turn, will enable you to vastly increase the potential business uses and benefits of the product at your firm.

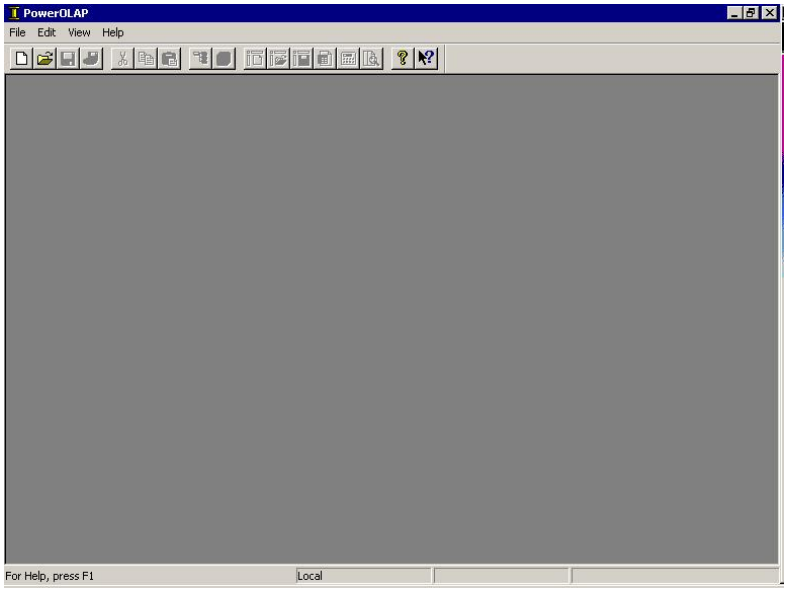
The *PowerOLAP User Manual*, in contrast to this manual, introduces PowerOLAP fundamentals by first allowing you to view and manipulate data within a pre-built model, then goes on to demonstrate how to perform and apply higher-level functions. That manual explains more comprehensively PowerOLAP's end-user features and functions. Therefore, please refer to the *PowerOLAP User Manual* for more detailed information on features discussed in this manual. Also, see the *OLAP Exchange*, *Synchronization Server*, or *MDB User Manual* for their respective features and functions, and how they are applied to create superior business solutions.

Creating a PowerOLAP Database

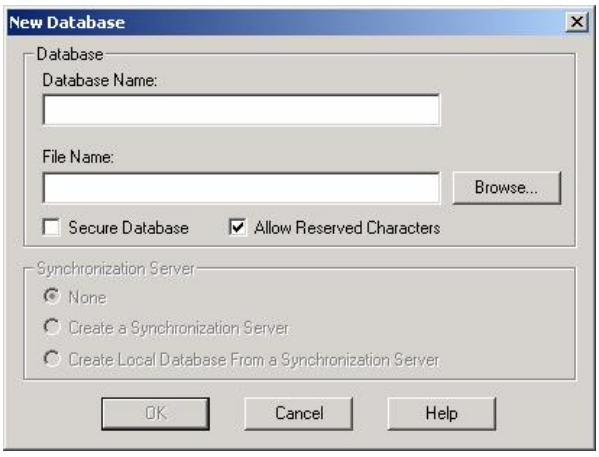
Create a New Database

Creating a PowerOLAP database is the first step in developing an application to store and model your data. The PowerOLAP database file, which has an ".olp" file extension, will contain all the components of your model. As you will see, these components include Dimensions and their Members; Cubes; Cube Formulas; and Slices, which display your data. Let's start by creating a new database, which you will name "QS Database" (short for Quick Start):

➲ From the Start Menu,
select **Start, Programs, PowerOLAP**.
The PowerOLAP main application window appears:

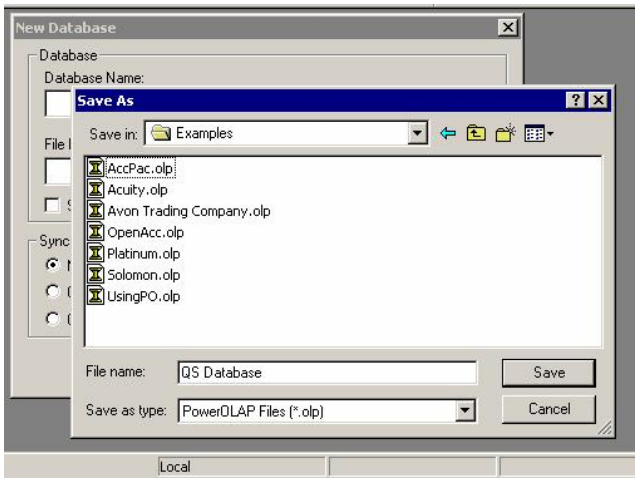


➲ Select **File, New Database**, or click on the New Database button on the toolbar. The following New Database dialog box is displayed:



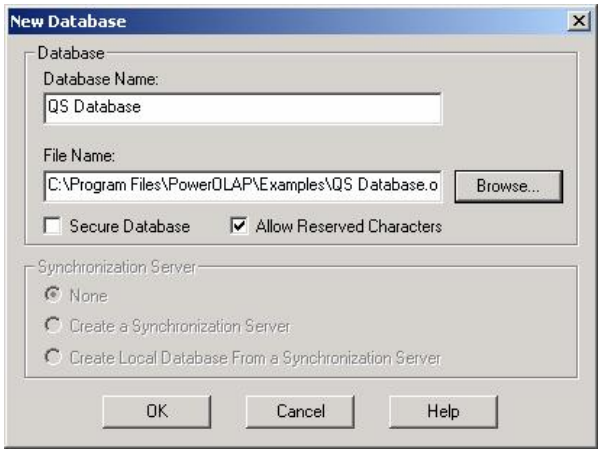
➲ Click **Browse** in the Database Name text box.

➲ In the Save As dialog, type **QS Database** in the File name text box, and follow the directory path
C:\Program Files\PowerOLAP\Examples.



➲ Click **Save**. The New Database dialog returns, displaying the Database Name as well as the path and file name of the database file.

In this case the Database Name will be the same as the File Name (shown in the following figure); you have the option to type in a different Database Name.



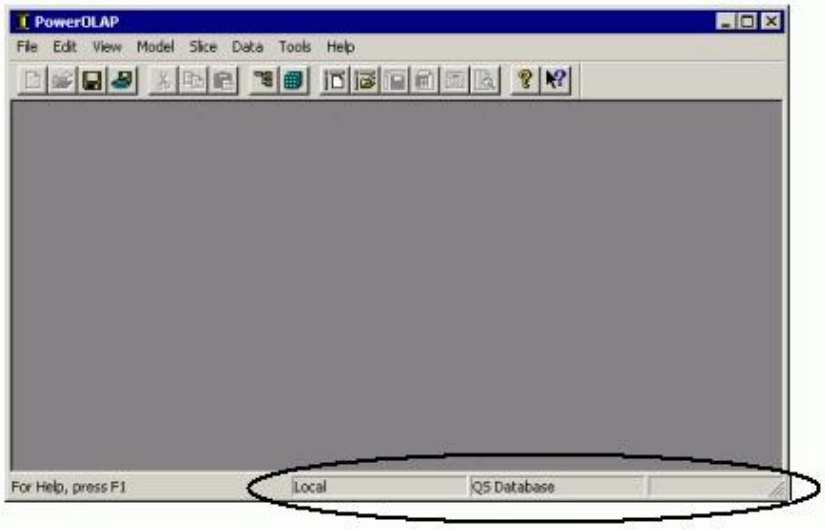
Notice the **Secure Database** and the **Allow Reserved Characters** checkboxes. Leave the default settings, unchecked and checked, respectively.

The **Secure Database** checkbox enables you to require a password to open the database. Thus if you check the box, then click OK, you will be prompted to give a password (and then verify it). See the *PowerOLAP User Manual*, Chapter 5, Applying Database Security, for a full explanation of Security features.

The **Allow Reserved Characters** checkbox allows you to use so-called “reserved characters”—e.g., double quote, period, comma, etc.—in your database. See General Options further on in this manual for a list of these characters.

The **Synchronization Server** area of the dialog refers to a PowerOLAP component that allows PowerOLAP databases to be synchronized via a shared file. The Synchronization Server area is activated if your license includes Synchronization Server capabilities (see the *Synchronization Server* manual); otherwise, it is grayed out (as above). Consult your Administrator to determine whether this tool is part of your application.

➲ Click **OK**. Note that in the PowerOLAP window, all eight menu items appear, and that several more toolbar buttons are now active.



Notice the status area, circled in the figure. From left to right, the boxes indicate:

Whether you are working in **Local** or **Server mode** (Server name will be indicated);

The **Database Name**; and

Synchronization Server Name, if active.

Only one database file (".olp") may be open at a time. Therefore, a new database can not be created if a database is currently open.

IMPORTANT: If you are working as a Client to an MDB Server, you cannot create a database on the Server. Typically, a new database shared by multiple clients would be created from the MDB Server Control Program. It is worth noting, though, that a user can create a database in standalone mode that can then be made available to the Server.

Creating Dimensions

Create a New Dimension

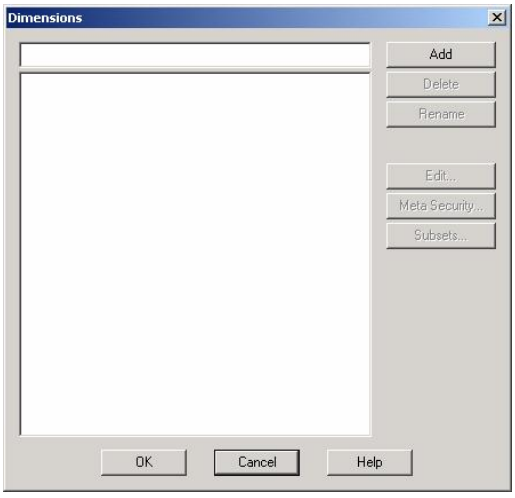
Dimensions are lists of related names—known as Members—that are used to identify data stored within a PowerOLAP database. These lists include names that you would see as headings across rows and down columns in a spreadsheet or standard report—e.g., *January*, *February*, *March*, etc. or the names of salespeople or regions in your organization. (If you are using OLAP Exchange to integrate data from a relational database application—like an accounting system—your structure of accounts would supply Members for the *Accounts* dimension: see the *OLAP Exchange* manual.)

Dimensions, then, are lists of related terms used to organize your data. Thus, a natural Dimension name for the members *January*, *February* and *March* might be *Months*. Dimensions, in turn, are used to construct Cubes, the multidimensional structures in which you store and model data.

In the model we are about to create, we will define three Dimensions: *Months*, *Accounts*, and *Regions*.

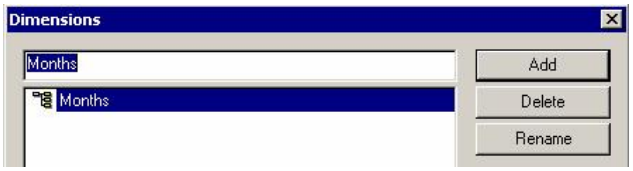
Create the *Months*, *Accounts*, and *Regions* dimensions as follows:

➲ Select **Model, Dimensions**. The following Dimension dialog is displayed.



➲ Type **Months** in the Dimension text box at the top of the dialog.

➲ Click the **Add** button.
The dialog appears as in the following figure:



➲ Type **Accounts** in the Dimensions text box.

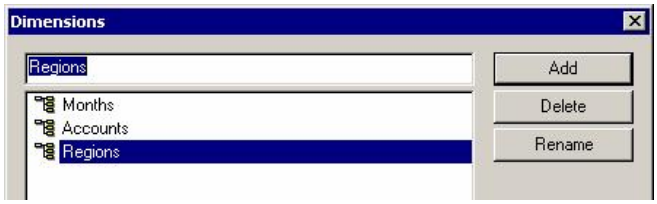
➲ Click the **Add** button.

➲ Type **Regions** in the Dimensions text box.

➲ Click the **Add** button.

You can use the **Enter** key in place of clicking the Add button to add Dimensions to the database.

Once you have entered the all of the Dimensions above, the list box will appear as in the following figure



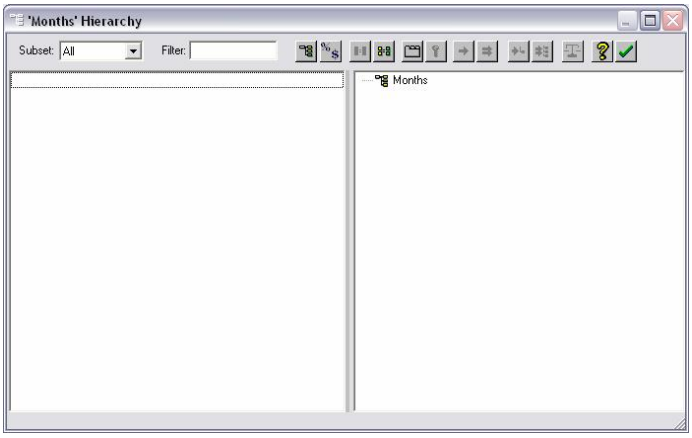
➲ Click **OK**. You are returned to the PowerOLAP main application window.

Adding Members to Dimensions

Create New Member

Dimensions are composed of Detail and Aggregate member types. Detail members "add up" to Aggregate members. For example, in the *Months* dimension you would make *January*, *February, March* (all Detail members) add up to *1st Quarter* (Aggregate member).
To add Members to a Dimension:

- Select **Model, Dimensions**. You are returned to the Dimensions dialog.
- Double-click on **Months**;
  or select **Months**, then click the **Edit** button.



The *Months* dimension is currently selected for editing, as indicated in the Dimension Hierarchy dialog title bar ('Months' Hierarchy).

Create New Member

- Select the **Create New Member** icon;
  or **right-click** within the Member list box (on the left in the dialog) and select **New**.
- Type **January** in the Members text box (over the currently highlighted text—e.g., Untitled3).
- Press **Ctrl-Enter**.
- Type **February** in the Members text box.
- Press **Ctrl-Enter**.
  The Member list box will appear as follows:

- Using this procedure, enter the **remaining months** of the year—these will be the Detail members in the *Months* dimension.

Next, you will create what will eventually be the Aggregate members for the *Months* dimension in the same manner (we say "eventually" because before a Hierarchy is created, all Members appear with the Detail icon—the number sign):

- Press **Ctrl-Enter**.
- Type **1st Quarter** in the Members text box.
- Press **Ctrl-Enter**.
- Type **2nd Quarter** in the Members: text box.
- Press **Ctrl-Enter**.
- Complete the *Months* dimension by adding **3rd Quarter, 4th Quarter** and **Total Year**.

You can double click on a name, or use the right mouse button, to rename or delete a name in the Member list box.

When you are done, the Member list box will appear as follows:

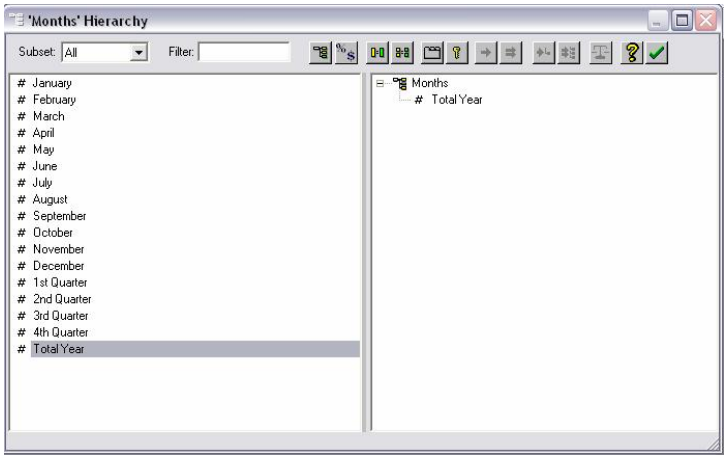
If the Member names are not in the order you want, click and drag them to the desired position in the list.

Proceed to the next page for the next step in Dimension design.

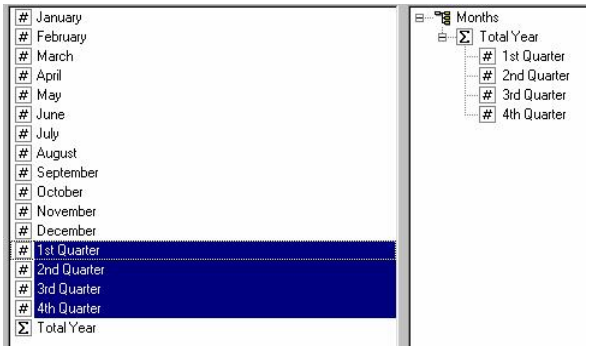
Creating a Dimension Hierarchy

Once the Members have been defined for a Dimension, the Hierarchy of these Members must also be defined. The Hierarchy determines the aggregation of Dimension members. You will now create the Hierarchy by selecting Members and moving them to the Hierarchy Definition box on the right side of the Dimension Hierarchy dialog. We will proceed by creating the Hierarchy for the *Months* dimension.
Define the Hierarchy for the *Months* dimension as follows:

- Select **Total Year** from the Members list box on the left.
- Drag it so that it is placed just below **Months** in the Hierarchy Definition box on the right, so that it appears as below:

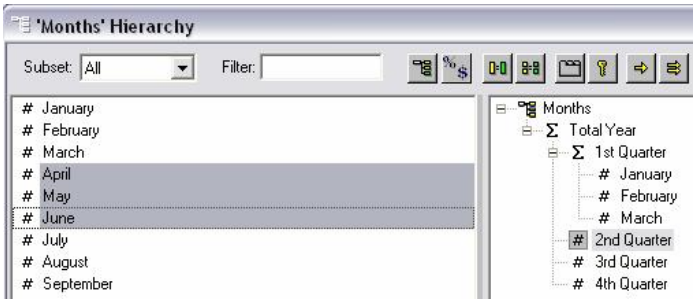


- Select **1st Quarter**, press the **Ctrl** key, and holding it down, select **2nd Quarter**, **3rd Quarter**, and **4th Quarter**.
- Place the selection just below **Total Year** in the Hierarchy Definition box, so that a sigma sign (for sum), Σ , appears beside **Total Year**. This indicates that Total Year is now an Aggregate member, the sum of the Members you placed below it (see following figure).



- Select **January**, hold down the **Shift** key, then select **March** in the Members list box so that **February** is highlighted also.

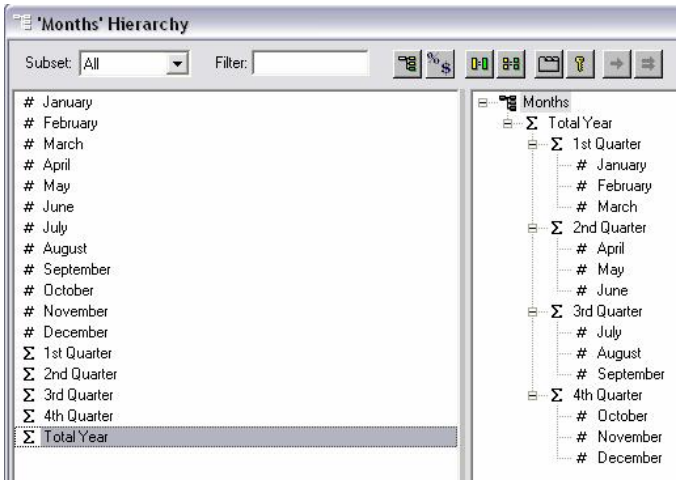
- Drag and place the selection just below **1st Quarter** in the Hierarchy Definition box. Now the sigma sign appears beside **1st Quarter**.
- Pressing the **Ctrl** key, select **April**, **May** and **June**. Release the **Ctl** key. Now Select **2nd Quarter** in the Hierarchy Definition box.



- Press the **Add Selected Members as Child** button, , on the toolbar. Notice that the sigma sign appears, next to 2nd Quarter.
- Bring over **July** as a Detail member under **3rd Quarter**.
- Highlight **August** and **September** from the list on the left and highlight **July** on the right.
- Click on the **Add Selected As Sibling** button, , to add August and September into the Hierarchy under the 3rd Quarter. Now August and September are shown under July, and the three Members comprise 3rd Quarter.

[Note that other buttons are available for Hierarchy creation: **Add all Members as Child**, , and **Add All As Sibling** button, .]

- **Complete the Hierarchy** for the 4th Quarter so that when you are done, the Hierarchy looks as follows:



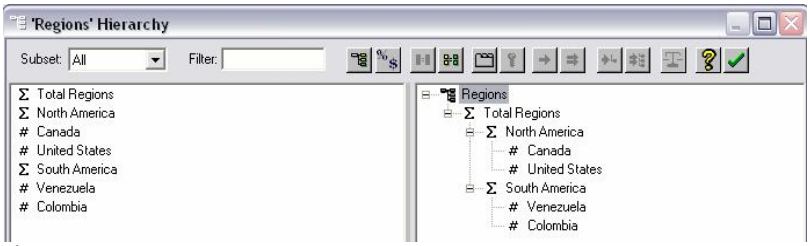
➲ Close the Dimension Hierarchy dialog by clicking the **OK** button, , the rightmost button on the toolbar.
You are returned to the Dimension dialog box.

Next, you will create the Hierarchy for the *Accounts* and *Regions* dimensions:
Add the following Members into the appropriate Dimensions. You will define their type, whether Detail or Aggregate, according to the Hierarchies you see in the following two figures (next page):

Dimension	Member	Type
Accounts	Net Sales	Detail
	Cost of Sales	Detail
	Gross Profit	Aggregate
Regions	Canada	Detail
	United States	Detail
	Venezuela	Detail
	Colombia	Detail
	Total Regions	Aggregate
	North America	Aggregate
	South America	Aggregate

When complete, the Dimension Hierarchies for Accounts and Regions should look as follows:

'Accounts' Hierarchy dialog box

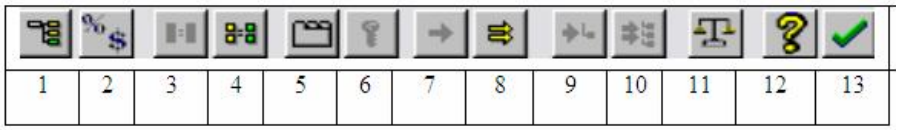


'Regions' Hierarchy dialog box

A hierarchical relationships—as you completed in the three Dimensions, *Months*, *Accounts* and *Regions*—defines a *parent-child* relationship between Members. Just as we have member *siblings* that exist on the same level in a Hierarchy under an Aggregate member, we also speak in terms of *child* members, which are defined as all Members that make up *parent* aggregations. It is important to note that a child member is not necessarily a Detail member. Child members may themselves be parents of other Members within a Hierarchy. For example, in the *Regions* dimension, North America is an Aggregate member; but it is also a child of *Total Regions*.

The Dimension Hierarchy Toolbar

The buttons on the toolbar in the Dimension Hierarchy dialog, and the functions they perform, are as follows:



1. **Create New Member**: Add a new Member (or press Ctrl-Enter to successively add new Members).
2. **Format**: Select formatting option for the corresponding cells.
3. **Member Aliases**: Define and edit Aliases and Alias Groups for the selected Member.
4. **Alias Group**: Add or delete Alias Groups, and assign names for each Member.
5. **Properties**: Assign Members a property for annotation purposes. A Member's properties can be displayed on an Excel worksheet when needed.
6. **Mark Member as Persistent**: Tag a Member as 'persistent' to prevent its data from being overwritten when updates or Cube re-builds occur.
7. **Add Selected as Sibling**: Insert the selection in the Member list on the left into the Hierarchy Definition on the right, below the selected Member.
8. **Add All as Sibling**: Insert all Members in the Member list after the selected Member in the Hierarchy, making them sibling Members.
9. **Add Selected Members as Child**: Insert selected Member(s) in the Member list as children of selected Member in the Hierarchy.
10. **Add All Members as Child**: Insert all Members in the Member list as children of selected Member in the Hierarchy.
11. **Specify Member Weights**: Edit the weight of a child Member (active only when a child Member is selected in the right hand Hierarchy Definition box).
12. **Help**: Accesses the help screen for the Dimension Hierarchy.
13. **OK**: Exits the dialog, with changes made and saved for the Dimension.

The functions listed above are described more fully in the *PowerOLAP User Manual*, though we will now discuss Aggregate weights in order to complete our QS Database dimension set-up.

Aggregate Weights

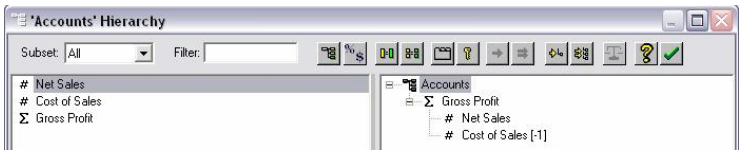
Specify Member Weight

Aggregate weights are used when a Member must be assigned a multiplied value in an aggregation in order for the data to be properly represented. For example, in the *Accounts* hierarchy, *Cost of Sales* should be handled as a negative number in the *Gross Profit* aggregation because it will represent an amount subtracted from *Gross Profit* (i.e., *Gross Profit = Net Sales - Cost of Sales*). To designate *Cost of Sales* as a negative number in the *Gross Profit* aggregation, we can assign it a weight of "–1". This means that the data included in *Cost of Sales* will always be multiplied by –1 when it is rolled up in the Hierarchy.

To "weight" *Cost of Sales*, complete the following process:

➲ Return to 'Accounts' Hierarchy dialog box (double-click on **Accounts** in the Dimensions dialog), and double-click **Cost of Sales** in the Hierarchy Definition box on the right. Note that, to the right of *Cost of Sales*, a box appears where "1" is highlighted and where you can enter a weight for the Member.

➲ Enter **-1** in the box so that it appears as follows:



➲ Click the **OK** button, , on the toolbar when complete.
You are returned to the Dimension dialog.

➲ Click **OK**. You are returned to the main application window.

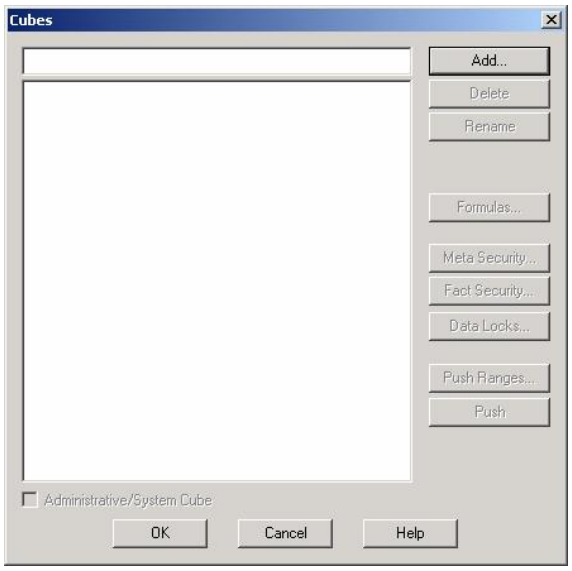
In PowerOLAP, the default aggregate weight is equal to '1'. Thus, a parent Aggregate member is simply the sum of all Child members defined in the Dimension hierarchy. In the example exercise, all aggregates you defined are standard with the exception of *Gross Profit*, in the *Accounts* dimension. Therefore, you do not need to define aggregate weights for the remaining Aggregate members.

Creating a Cube

Cubes

Using the Dimensions created in the previous exercises you will now create a PowerOLAP Cube that will store and model your data.
To create a *Current Year Budget* cube:

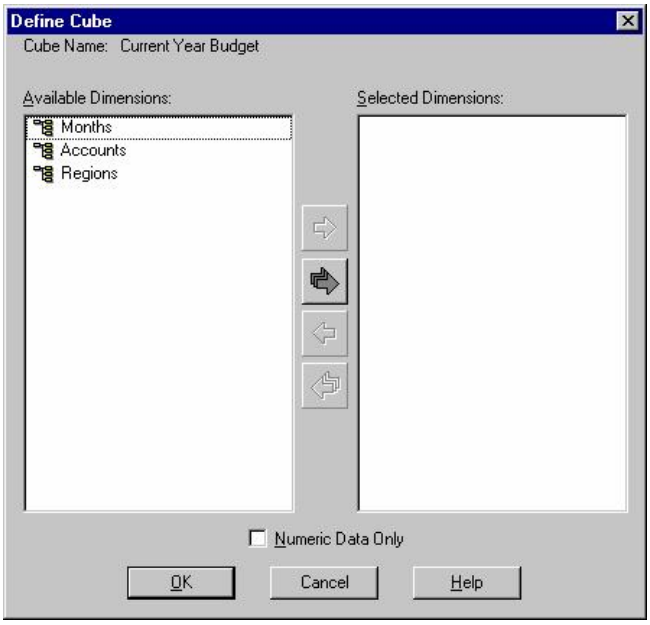
➲ From the main application window, select **Model, Cubes**.
The Cubes dialog is displayed:



➲ Type **Current Year Budget** in the Cubes text box.

➲ Click the **Add** button.

The Define Cube dialog appears, in which you select Dimensions to be used by the Cube:



➲ Select all of the dimensions in the Available Dimensions list box by clicking the button.

All three Dimensions are moved to the Selected Dimensions list box on the right.

➲ Click **OK**. Note that the *Current Year Budget* cube is now listed in the Cubes dialog.

All of the Cubes dialog's buttons on the right are now activated. These buttons control functionality associated with Formulas and setting Security privileges, as well as the OLAP Exchange capability to push data ranges back to a relational database—they are covered in depth in the *PowerOLAP User Manual* and *OLAP Exchange* manual respectively.

➲ Click **OK** to return to the main application window.

The *Current Year Budget* cube is now ready for data input.

Creating a Slice View

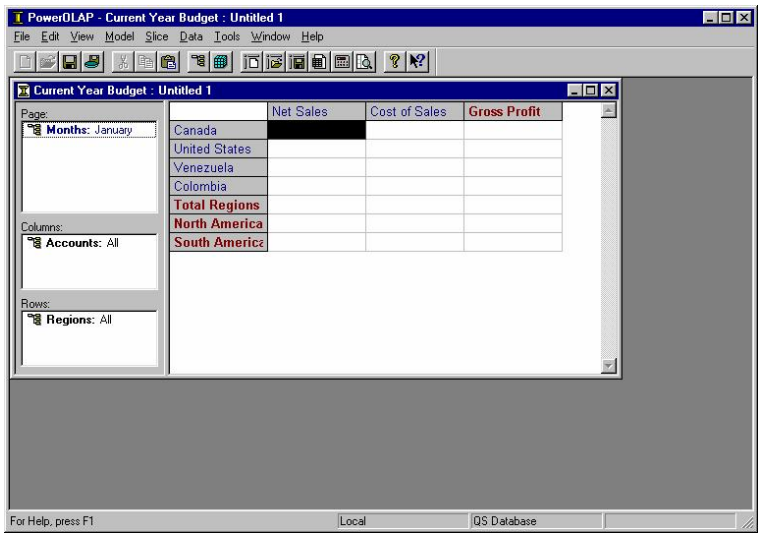
New Slice

PowerOLAP provides a method for looking into a Cube to view and input data. This means of viewing and inputting data is known as "creating a Slice." A Slice is a two-dimensional view of a Cube that arranges data in a grid, just as a spreadsheet does. You can create slices "on the fly" to see any view of a Cube, or you can save and re-open Slices for ongoing data viewing or inputting. Finally, as you will see, you can instantaneously create an Excel spreadsheet from any Slice view.

To create a Slice called *Regions by Account:*

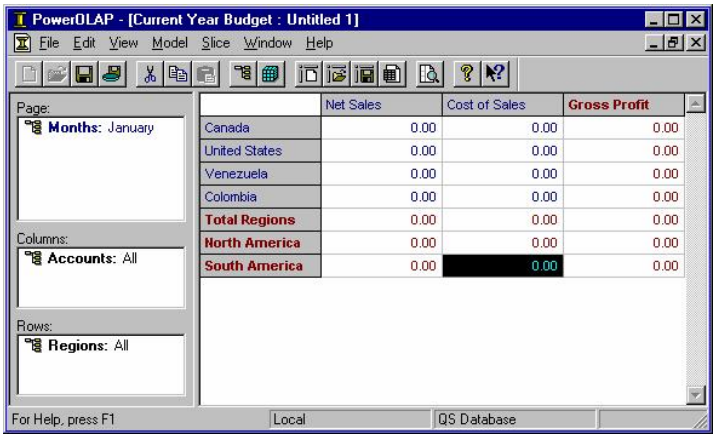
➲ Select **Slice, New**. The following New Slice dialog box is opened. The list box displays the names of available cubes in your database. In our case, we just created the only cube listed, *Current Year Budget*.

➲ With the *Current Year Budget* cube selected, click **OK**.



The Slice View dialog is opened in the Content Area of the main application window, as in the above diagram. The window displays a *Current Year Budget* slice, as yet untitled, and with no data in the grid.

➲ Press **F9**. (This manually recalculates the grid's data, explained further below.) Keep in mind that, as yet, no figures have been entered into the Cube, so you will see zeros as data throughout the Slice.



By default, when PowerOLAP creates a new Slice, it places the last Dimension brought into the cube when it was created in the Rows position, the next-to-last Dimension in the Columns position, and any remaining Dimension(s) in the Page position. (When we created the Cube in the last exercise, we brought all Dimensions into the cube at once in the order they were listed.) In the above example Slice, *Accounts* are displayed as columns, *Regions* as rows and *Months* as the *Page* dimension, currently displaying *January*, which is the first Member entered for the *Months* dimension.

➲ Select **Slice, Save As** to save the Slice.
The following dialog is displayed:

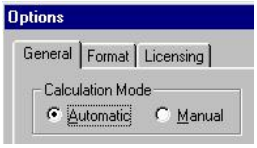
Save Slice

➲ Type **Regions by Accounts** in the Slice Name text box.

➲ Click **OK**. You are returned to the Slice—note that both the Cube name, *Current Year Budget*, followed by the Slice name, *Regions by Accounts*, appear in the title bar.

NOTE: PowerOLAP's default calculation mode is set to Manual. Thus, when you make changes to a Slice, you will need to Press F9 (or the calculator button on the menu bar) to see those changes reflected in the Slice. You can change the calculation mode to Automatic by selecting **Edit, Options**, and clicking on the **Automatic** radio button in the General tab.

➲ Change the calculation mode from Manual to **Automatic** on the Edit menu under Options.



Now you will see changes instantly on screen as they are made throughout the remainder of this manual's exercises.

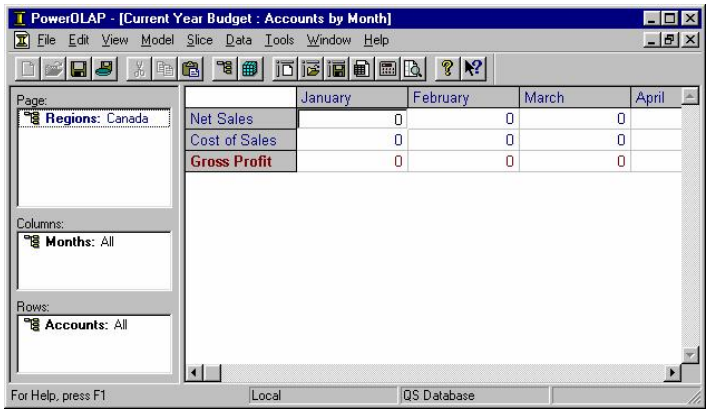
Arranging Slice Dimensions

To demonstrate how quickly and easily views can be changed, you will now arrange the Dimensions of this Slice to view data with *Accounts* as rows, *Months* as columns and *Regions* displayed as a page. Start by dragging and dropping the Dimension names into the appropriate list boxes:

- Select **Months** in the Page list box and drag it down to the Columns list box, below *Accounts*. [Note that a "nested" view is created, assuming you are operating in Automatic calculation mode.]
- Select **Regions** in the Rows list box and drag it up to the Page list box.
- Select **Accounts** in the Columns list box and drag it down into the Rows list box.

By placing *Regions* in the Page list box, you display data for a single Member of the *Regions* dimension. The Page member you see when you first arrange a Slice is the Member at the top of that Dimension's member list. In this case, the Slice grid displays the data for all *Accounts* and all *Months* for the *Region*s member *Canada*.

- Select **Slice, Save As** and type **Accounts by Months** in the Slice Name text box.
- Click **OK**. You have created and saved a second Slice, *Accounts by Month*:



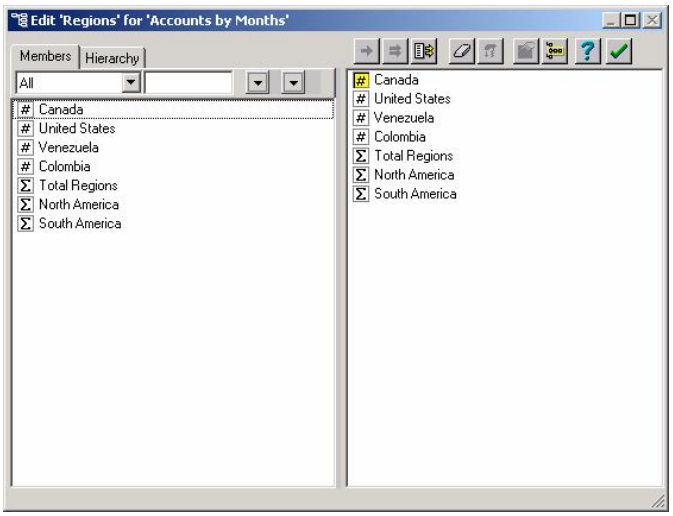
Selecting Page Members

Currently you are viewing data for *Canada*. To view data for other Members defined in the *Regions* dimension within the *Accounts by Months* slice—e.g., to change the view from *Canada* to *United States*:

➲ Double click on **Regions: Canada** in the Page list box.

Set Page Member

The following Edit Slice dialog is displayed (in this example, Edit 'Regions' for 'Accounts by Months'):



Note that the Detail member icon to the left of *Canada* in the Slice Content list box is yellow, indicating that this Member is the currently selected Page member.

You can select any Member in this list box as the Page member to view within the slice:

➲ Double-click **United States** in the Slice Content list.
The icon beside it is now yellow. You may also select a Page member by clicking on the Select Page Member icon, , from the menu bar.

➲ Click the **OK** button, , on the toolbar, to close the dialog and return to the *Accounts by Months* slice.

The current Slice now displays data for *United States*, which is indicated in the Page list box, beside *Regions* (i.e., *Regions: United States*).

Changing the Grid Layout

You can change the layout of the Slice grid by moving Members of a Dimension within the Slice Content list box:

➲ Double-click on the **Months** dimension in the Columns list box in your current Slice. The Edit Slice dialog box is displayed.

Currently, *Total Year* is at the bottom of the Slice Content list box, which corresponds to the rightmost column in the grid (you may need to scroll rightward in the grid to see *Total Year*). By dragging and dropping *Total Year* to the top of the list, you can move it to the leftmost column:

➲ Select **Total Year** from the Slice Content list box on the right.

➲ Drag and drop **Total Year** above **January**, as circled below.

➲ Click the **OK** button, , on the toolbar.

You are returned to the *Accounts by Months* slice, and *Total Year* is now displayed in the first column of the Slice.

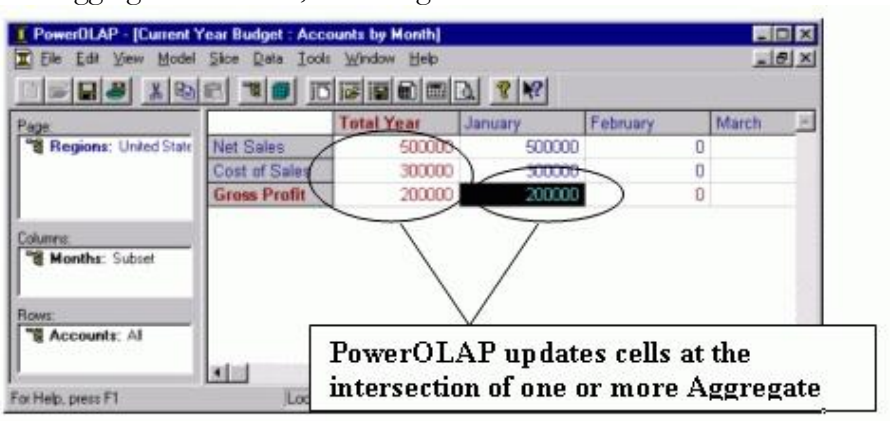
Entering Data in a Slice

So far, you have demonstrated PowerOLAP's remarkable flexibility in organizing and displaying data within a Slice. Next, you will demonstrate another key function of the Slice: using a Slice to enter data directly into the underlying PowerOLAP database.
Currently, the data in the *Current Year Budget* cube is all zeros because it is a new Cube and data has not yet been entered into it.
To enter data into the *Accounts by Months* slice:

➲ Click the cursor at the intersection of **January** and **Net Sales**.
Type **5000**, then press **Enter**.

➲ Click the cursor at the intersection of **January** and **Cost of Sales**, Type **3000**, then press **Enter**.

Notice that PowerOLAP has automatically adjusted the values in data cells that occur at the intersection of Aggregate members, reflecting the values entered above.



➲ Try to type **100000** at the intersection of **Total Year** and **Gross Profit**.

PowerOLAP does not allow you to change the values in data cells involving one or more Aggregate members. PowerOLAP will automatically update these cells only when the values in relevant Detail members change.

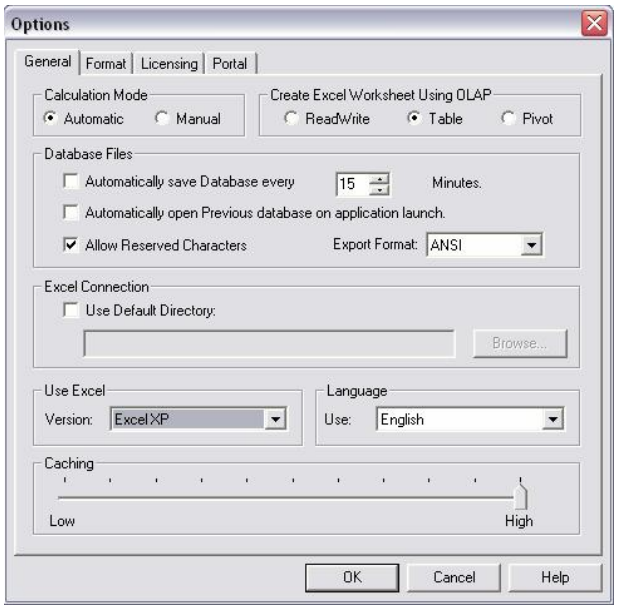
Setting General & Format Preferences

Before continuing to work with data in a Slice, we will take a look at some preferences, among which are those that affect the look of a Slice

General Tab

Selecting the Options command from the Edit menu enables you to set several general database options.

➲ Select **Edit, Options**. The Options dialog box is opened, with the **General tab** on current settings.



Calculation Mode

As discussed previously, the radio buttons will allow you to show Slice changes and data entry calculations upon entry (Automatic) or upon pressing F9—similar to your experience with Excel.

Create Excel Worksheet Using OLAP

Read/Write, Table or Pivot: These concern the data connection function used to create Excel spreadsheets from PowerOLAP slices. You will create an Excel worksheet momentarily—these functions are discussed in detail in the *PowerOLAP User Manual*.

Database Files

When Automatically save Database every is selected, you can save an active database periodically (every *X* minutes) by adjusting the spinner scrollbars or entering a number directly in the text box.

When Automatically open Previous database on application launch is selected, the last-opened database will open when PowerOLAP is opened.

When Allow Reserved Characters is checked, PowerOLAP can accept any of the following characters in Meta Data:

Double Quote	"
Single Quote	'
Period	.
Comma	,
Semi-colon	;
Brackets	[ ]
Braces	{ }
Parenthesis	( )

If this function is not checked, PowerOLAP will NOT accept these characters in the Meta Data. This becomes important when working with OLAP Exchange, as you will not be able to build a Cube if the source data contains any of these characters.

To Use Default Directory—i.e., set up the PowerOLAP database to default to a certain directory upon opening or saving a file:

➲ Select the **Use Default Directory** tick box. This option saves you from having to browse for the database when opening each time by defaulting to the folder you designate. Note: you must have the database closed when setting this option.

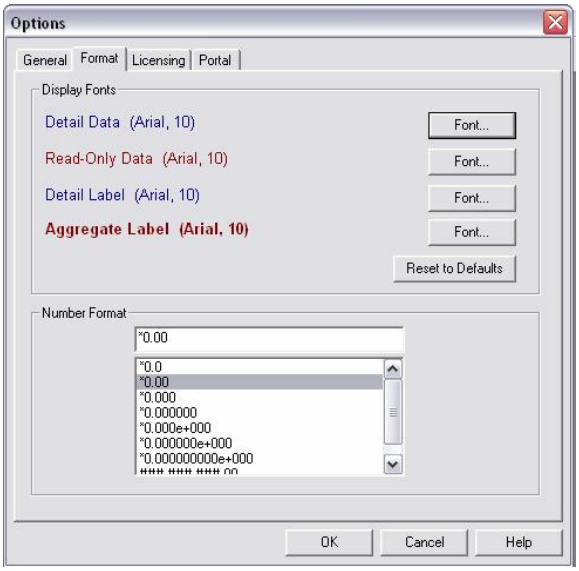
Use Excel (Version)
During installation, PowerOLAP prompted you for a version of Excel to use. You can select a different version of Excel to use here: **Excel 97**, **Excel 2000**, **XP, Excel 2003**, or **Latest Installed** version.

Language
Select your language preference here. PowerOLAP is available in the following languages: English, Danish, French, Portuguese. After selecting a language, you will be informed that in order for the selection to take effect, you must close and then re-open PowerOLAP.

Caching
The sliding bar determines the size of the cache threshold.
PowerOLAP stores aggregation and formula calculations in cache based on the amount of cache available. If the cache threshold is set to high, PowerOLAP will continue to store calculations in cache, hence decreasing the time to retrieve values. If you have a large amount of RAM on the Server machine, you can move the bar to the right.
If the cache threshold is set to low, PowerOLAP will store only a minimal amount of information in cache, and recalculations will be retrieved from the data store. Move the bar to the left, therefore, to limit the amount of cache devoted to calculations—if you have less RAM, this may be the better choice. You will continue to see calculations quickly, though not as fast as if they were retrieved from cache.

Format Tab

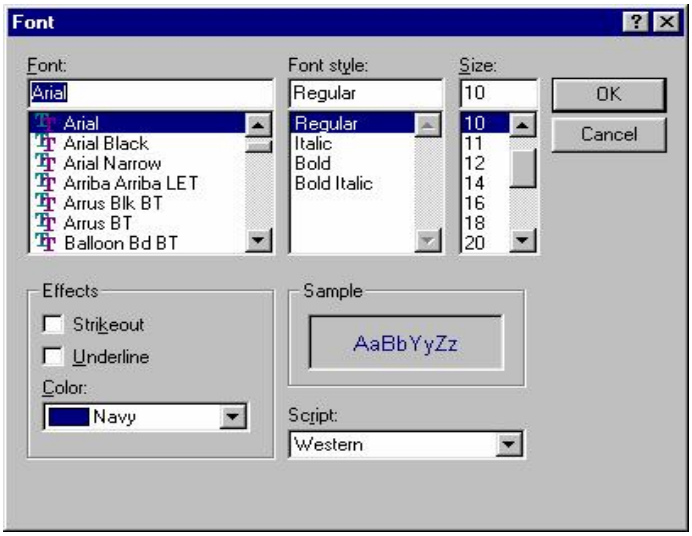
If you would like to make format changes to the numbers or the text (decimal places, colors, font size) as they appear in a Slice, choose the Format tab.



From here, you can adjust the appearance of the **Display Fonts** and the **Number Format** in PowerOLAP. (To change the appearance of these elements within Excel after creating a worksheet from a Slice—as will be discussed shortly—format the worksheet as you normally would).

For Display Font changes:

➲ Click on the **Font** button corresponding to the element whose appearance you wish to change. This brings up the Font dialog:



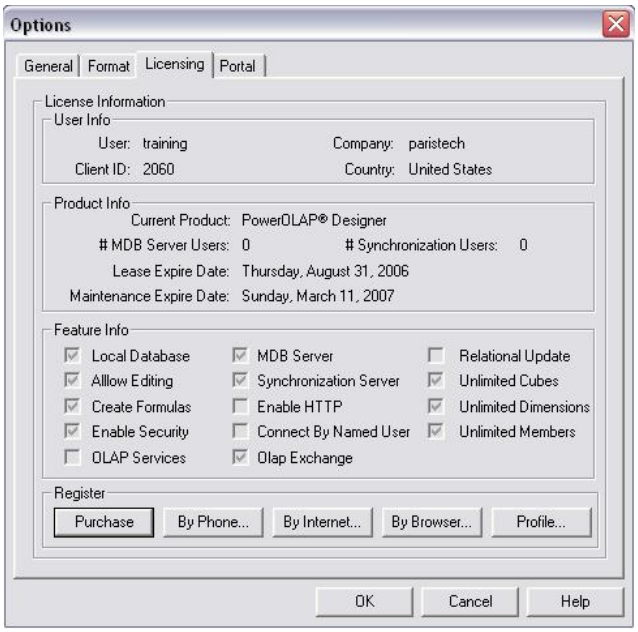
➲ Select the preferred **Font Type**, **Style**, **Size**, **Effects**, **Color** and **Script** for your database and click OK.

Number formatting refers to decimal places and commas in numbers.

➲ Highlight the **Number Format** you prefer and click **OK**.

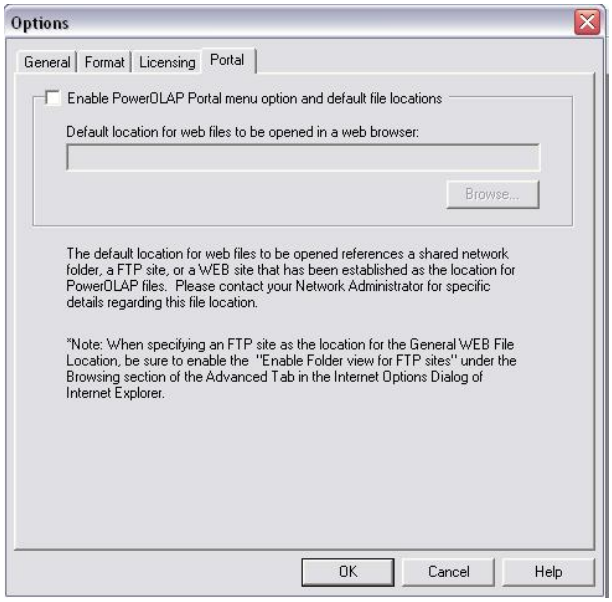
Licensing Tab

The Licensing Tab provides information about your current license, and the functionality available after activation. It also provides for registering and activation of your product.



Portal Tab

The Portal Tab provides information to enable and set up a Portal.


Options
General
Format
Licensing
Portal
Enable PowerOLAP Portal menu option and default file locations
Default location for web files to be opened in a web browser:
Browse...
The default location for web files to be opened references a shared network folder, a FTP site, or a WEB site that has been established as the location for PowerOLAP files. Please contact your Network Administrator for specific details regarding this file location.
*Note: When specifying an FTP site as the location for the General WEB File Location, be sure to enable the "Enable Folder view for FTP sites" under the Browsing section of the Advanced Tab in the Internet Options Dialog of Internet Explorer.
OK
Cancel
Help

Saving Changes to a Database

There are two options for saving a database. The Save Database command found on the File menu allows you to save an open database to disk or hard drive; however, this does not save modified new or modified Slices.

Save Database

To save the database with the current name:

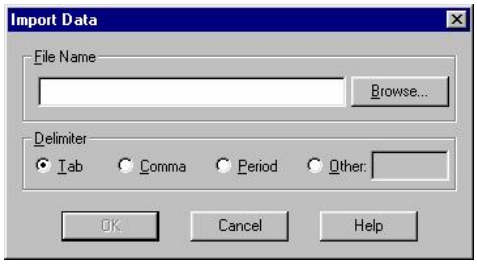
➲ Select **File, Save Database**. (To save the database under a new name select **File, Rename Database**)

PowerOLAP provides an additional Save command—Save All, which is represented on the Toolbar by the following button: . The Save All command will save the current database, all Slices, all entered data; and additionally, Save All will save any changes made to the Slice layout. Before choosing this option, decide if you want to save all your open Slices as they are currently configured.

Importing Data from a File

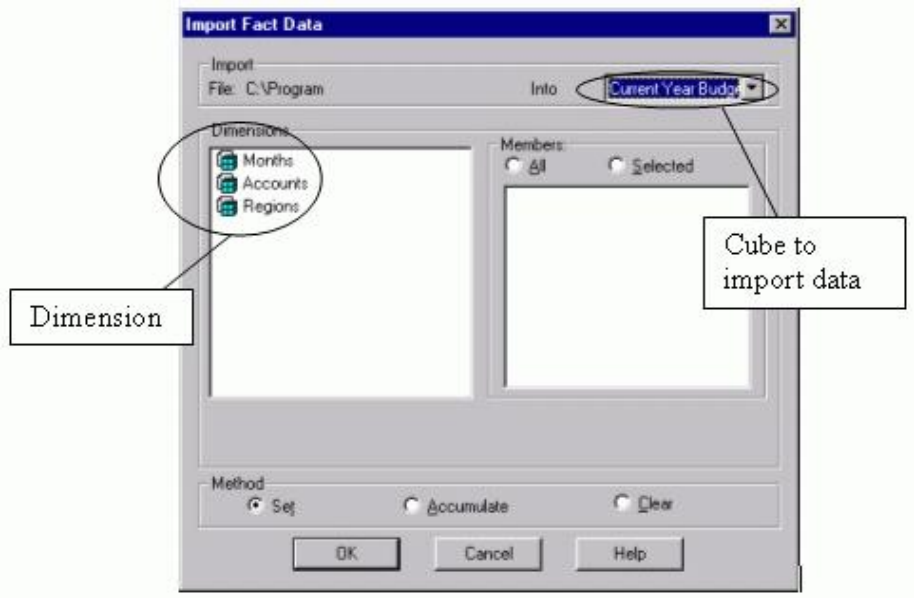
At this point in the example, you have entered and stored some data in the *Current Year Budget* cube for the *United States,* in *January*. The next exercise will populate the remaining detail *Regions* members by importing *January* data contained within a sample text file included with the PowerOLAP application. This file, *Qsdata.txt*, is located in the *C:\Program Files\PowerOLAP\Examples* subdirectory (assuming that the C drive is where your Program Files directory is located) created during the installation process.
[Note that importing text data, along with entering data via a Slice or an Excel worksheet are two methods bring data into PowerOLAP. Another method is to dynamically integrate Meta and Fact data from a relational database, via OLAP Exchange.]

- Select **Data, Import**.
- Choose **Fact Data**. The Import Data dialog box is displayed:



- Click **Browse** to select the file to import.
- In the Open dialog box, select the **QSData.txt** file from the …*PowerOLAP\Examples* subdirectory.
- Click **Open**. You are returned to the Import Data dialog.
  Keep Tab as the selected Delimiter.
- Click **OK**.

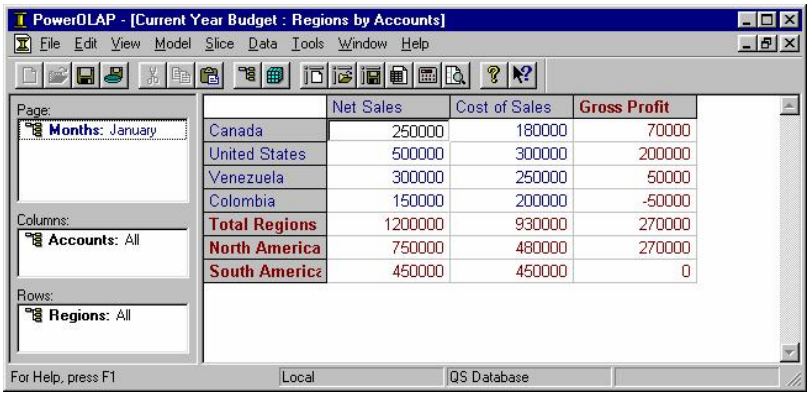
The Import Fact Data dialog is opened. This dialog box shows the Dimensions that appear in the text file as well as in the selected Cube—at the top right—into which you will import data. The *Current Year Budget* cube is selected.



- Click on the **Months** Dimension.
  A message box appears stating that the selected Dimension appears in both the import file and the Cube. Also, it states that showing common Members could be time-consuming (here, it will not).
- Click **Yes** to confirm that you want to continue.
  Note that, in the Members list box, only *January* is selected for this Dimension; this is because the import file contains data for *January* only. If you click on the *Accounts* and *Regions* dimensions you will see that the import file contains data for several of their Members—but the data will only be for the month of *January*.
- Click **OK** to begin the import process. You will receive the following prompt indicating that the import was successful:

- Click **OK**. You are returned to the *Accounts by Months* slice.
- Select **Slice, Open** to access the Open Slice dialog.
- Double-click on **Regions by Accounts** within the Slice list box.

The *Regions by Accounts* slice now displays the data just imported.



The import file populated *Net Sales* and *Cost of Sales* data for *January* for all Members of the *Regions* dimension. PowerOLAP automatically calculated the cells at the intersection of Aggregate members (for example, *Net Sales* for *Total Regions*, etc.).

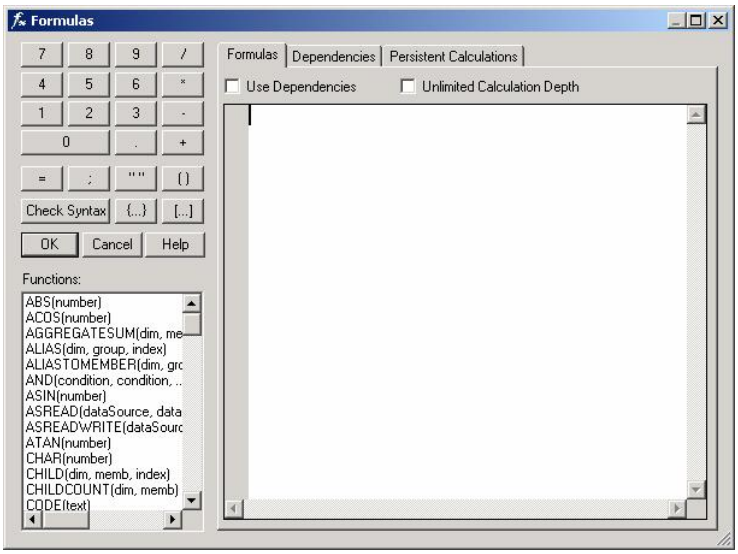
➲ Select **File, Save All** to save changes you made both to data and to slice views.

➲ Close the *Regions by Accounts* slice, leaving open the *Accounts by Months* slice.

Creating Cube Formulas

You have seen how creating Dimension hierarchies, and assigning Aggregate weights, results in the natural "summing up" of values. Cube formulas represent a powerful extension of your ability to perform mathematical calculations throughout a PowerOLAP database. With Cube formulas, you can perform all manner of calculations to populate a cell, ranges of cells, even entirely different Cubes.
Presently your *Current Year Budget* cube contains data for *January* for all *Accounts* and *Regions*. The following Cube formula will populate the month *February*.

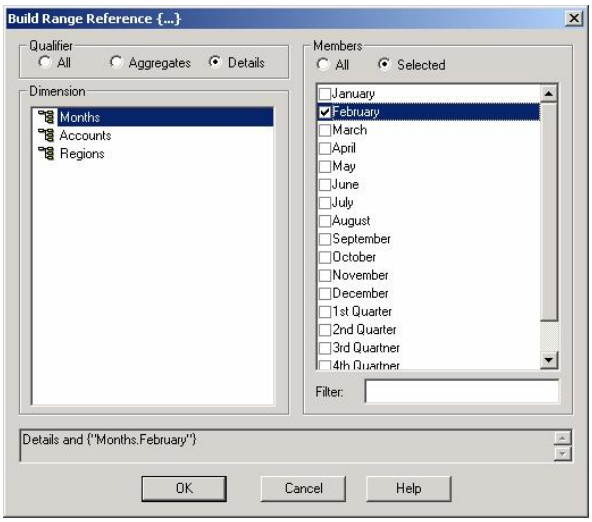
- Select **Model, Cube**. The Cube dialog box is opened
- Click on the **Current Year Budget** cube to select it.
- Click on the **Formulas** button. The Formulas dialog is opened:



We will now make use of some buttons in the Formula dialog (also known as the Formula Editor). These buttons will enable us to specify the area of the cube we want to populate with data, and where the data will come from.

- Click on the "**squiggly brackets**"— {...} .
  The Build Range Reference dialog appears.
  Months is selected in this dialog at the top of the Dimension list; it is the Dimension we want to work with, so leave as is.
- For the Qualifier (top left), select the radio button **Details**.
  This indicates that only Detail members are to be calculated by the Cube formula— Aggregate members will be calculated according to the Dimension hierarchy. (Note: it is possible to "overwrite" hierarchy calculations via a Cube formula, a very important feature if you wish to calculate a "what if" or budgeted Aggregate data point, so that it contrasts to actual figures in Detail data points.)

➲ On the right, select **February** among the Months.
(The Selected radio button is selected, as a consequence.)
The dialog appears as follows, with the formula as it exists so far, at the bottom:



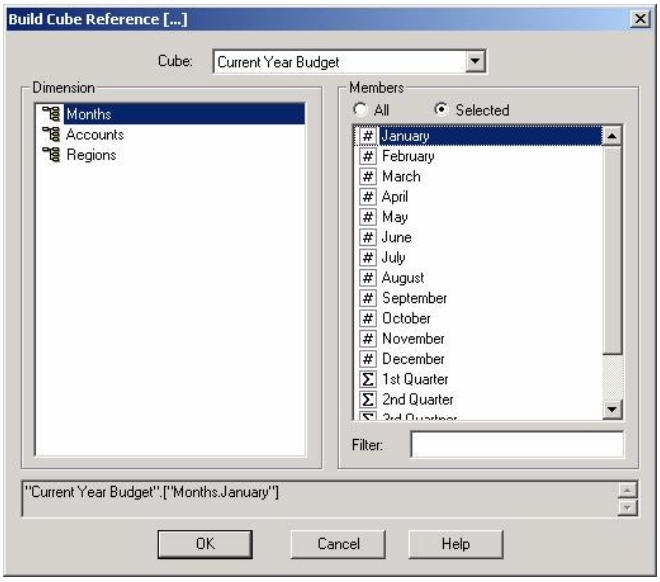
➲ Click **OK**.

➲ Click on the "equals" sign in the Formula Editor, [=].
The left-hand side of the formula is completed, and is shown in the content area.

```
All and {"Months.February"}=
```

➲ Next click on the "**square" brackets** — [...] .
The Build Cube Reference dialog appears.
Again, **Months** is selected; it is the Dimension we want to work with, so leave as is.

➲ Select **January** from the Member list on the right.
(The Selected radio button is selected, as a consequence.)
[Note that at the top of the dialog, there is a Cubes drop-down.
This points up an important feature—the ability to create cross-cube formulas, which is explained in the *PowerOLAP User Manual*. There is only one cube in our database, *Current Year Budget*, in the formula we are creating, data will come from this cube, to populate another area of the same cube.]
The Build Cube Reference dialog appears as follows:



➲ Click **OK**.
The Formula Editor content area appears as follows (you can hit Enter after the "=" to show the formula on two lines)"

```
All and {"Months.February"}=
"Current Year Budget".["Months.January"]
```

➲ Use the buttons in the Formula Editor — the asterisk (for multiplication), the numbers and the semi-colon — to complete the formula to that it appears like so:

```
All and {"Months.February"}=
"Current Year Budget".["Months.January"]*1.5;
```

Following is a breakdown of the syntax of the Cube formula:

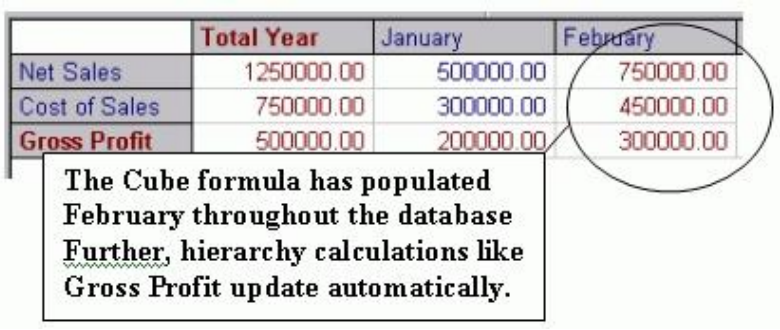
Left of equal '='	Area of cube to populate
Right of equal '='	Formula
{"Months.February"}	Dimension and Member to populate
"Current Year Budget"	Source cube
["Months.January"]	Range within source cube
1.5	Value (in this case, +50%)
;	Ends formula statement

➲ Click **OK** to save the formula.
(If you have mistyped the formula, you will receive a message indicating that there is syntax problem.)
You are returned to the Cubes dialog box.

➲ Click **OK**.

➲ Press **F9** inn the *Accounts by Month* slice to recalculate values.

Notice that the February column has been populated by the Cube formula defined in the previous steps.



Next, you will create a Cube formula that calculates a ratio of two Members. You will first need to add a new Member—*Margin %*—to the *Accounts* dimension, and then modify the *Accounts* dimension hierarchy. This Cube formula exercise points up two important strengths of PowerOLAP, in comparison to static modeling tools, OLAP or otherwise: the capability to create new, "on-the-fly" calculations (which can of course be subsequently saved) for precisely specified (even new) components of a business model, which themselves are *created entirely within PowerOLAP* [i.e., not dependent on any static model of business data].

➲ Select **Model, Dimension**.

➲ Double click **Accounts** in the Dimension list box.
The 'Accounts' Hierarchy dialog box is displayed.

➲ Click on the **Create New Member** button, , on the toolbar.

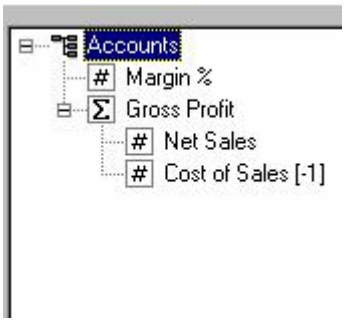
➲ Type **Margin %** so that it appears in the Members list box..

To modify the *Accounts* dimension hierarchy:

➲ Expand **Accounts** in the Hierarchy Definition box, on the right.

➲ Select **Margin %** from the Members list box and drag it to the Hierarchy list box and release it just under *Accounts*.

The completed hierarchy will appear as in the following figure:



➲ Click the **OK** close the Dimension Hierarchy dialog.

➲ Click **OK** in the Dimensions dialog box.
Margin % now appears as the top row in the *Accounts by Month* slice.

Next, you will define a Cube formula that creates values for the new *Margin %* : the values will be based on a formula that divides *Gross Profit* by *Net Sales*:

➲ Select **Model, Cube**.

➲ Double-click the **Current Year Budget** cube.

➲ Press **Enter** twice in the Content Area to move the previously written formula down two lines. Begin on the top line of the Formula Content area, *i.e., place this formula above the other*.

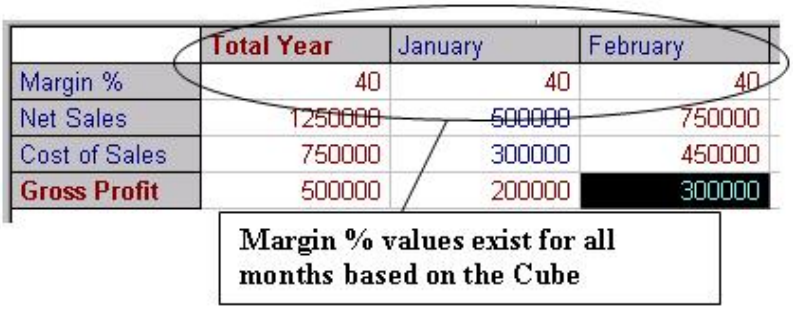
Priority, which is top-to-bottom in the Formula editor, is very important for determining data calculations—consult the *PowerOLAP User Manual.*

➲ Using the Formula Editor dialog, create the formula:
All and {"Accounts.Margin %"}=
"Current Year Budget".["Accounts.Gross Profit"]/
"Current Year Budget".["Accounts.Net Sales"]*100;

➲ Click **OK** to save the formula.
You are returned to the Cubes dialog box.

➲ Click **OK**.

➲ Press **F9** to recalculate values in the *Accounts by Month* slice.



Margin % is now calculated for all *Months* in the *Current Year Budget* cube. The figure above shows the *Margin %* figures for *United States*. You can select other *Regions* members to verify that all members in the *Regions* dimension have been updated as well.

➲ Select **File, Save All** to save the data and the Slice (which now includes *Margin %*) to disk.

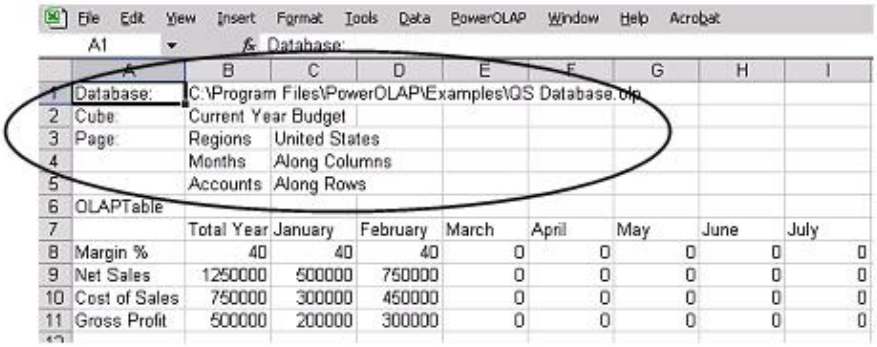
Creating an Excel Worksheet

Create Excel Worksheet

One of PowerOLAP's key features is that it enables you to create an Excel worksheet from a PowerOLAP slice. You can then work with data in Excel, utilizing all that product's features and functions while maintaining a dynamic connection to the PowerOLAP database. This is why PowerOLAP is credited with having a "spreadsheet front end."
To create an Excel worksheet from the *Accounts by Months* slice:

➲ Select **Slice, Worksheet**; or press **F8**; or click on the create worksheet button, , from the menu bar.

PowerOLAP launches Excel (assuming it is not running), displaying the newly created worksheet. A new Excel worksheet appears, as follows:



	A	B	C	D	E	F	G	H	I
1	Database:	C:\Program Files\PowerOLAP\Examples\QS Database.olp							
2	Cube:	Current Year Budget							
3	Page:	Regions	United States						
4		Months	Along Columns						
5		Accounts	Along Rows						
6	OLAPTable								
7		Total Year	January	February	March	April	May	June	July
8	Margin %	40	40	40	0	0	0	0	0
9	Net Sales	1250000	500000	750000	0	0	0	0	0
10	Cost of Sales	750000	300000	450000	0	0	0	0	0
11	Gross Profit	500000	200000	300000	0	0	0	0	0

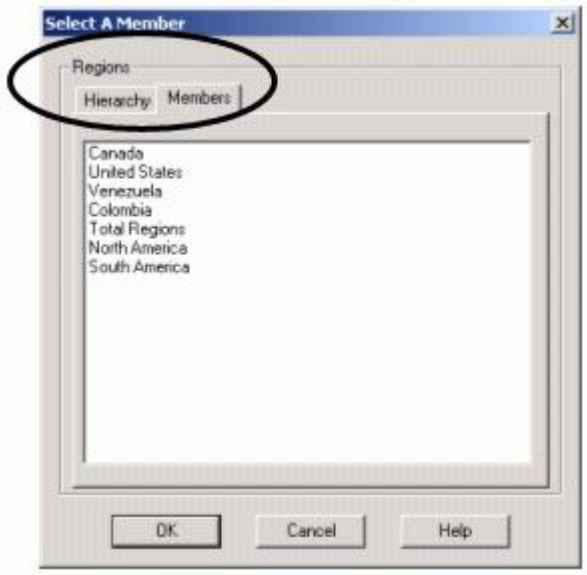
The first few rows of the worksheet display information indicating the PowerOLAP database; the Cube; the Page Dimension member(s) that the Slice data shows (in the figure above, one Page Dimension, *Regions*, and *United States* is shown); and the Dimensions "Along Rows" and "Along Columns".
The worksheet can now be saved as an XLS file via Excel's **Save** command.

Selecting a Page Member to View in Excel

Change the Page member in Excel as follows:

➲ Double-click Page member cell—e.g., cell **C3** (United States).

The Select a Member dialog box appears. Note the two tabs (circled): you can find Members based on where they appear in the dimensional hierarchy or in the Member list (this tab is selected below).



➲ Select **Canada**.

➲ Click **OK**, and then press **F9** to update the worksheet.

The worksheet now shows data for the new Page Member, *Canada*. You can repeat this means of selection—via the Select a Member dialog, which PowerOLAP has made available in Excel—to pick other countries in the *Regions* dimension. [Were this a four-, five- , etc. dimensional cube, you could pick any number of Page members to view, multiplying your potential sheaf of reports manyfold!]

Entering Data from within Excel

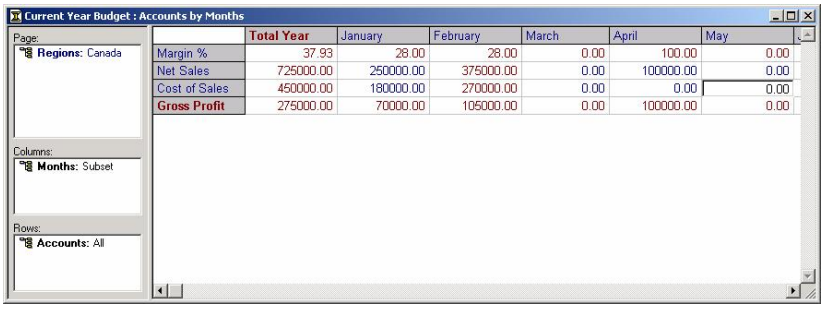
You can enter data into a PowerOLAP database using an Excel worksheet. This has great applicability in forecasting, planning and budgeting systems that use PowerOLAP. All data entered into a worksheet is automatically updated using one of PowerOLAP's functions (OLAPTable has been shown here), each of which maintains a "bi-directional, dynamic spreadsheet connection" between PowerOLAP and Excel.
To enter data via an Excel worksheet:

- Select a cell at the intersection of detail members, such as **F7**, which is the cell at the intersection of *April, Net Sales*.
- Type **100000**.
- Press **Enter**, and then press **F9** (if Excel is set to Manual calculation) to recalculate the worksheet. The worksheet appears as below:

	A	B	C	D	E	F	G
1	Database:	C:\Program Files\PowerOLAP\Examples\QS Database.olp					
2	Cube:	Current Year Budget					
3	Page:	Regions	Canada				
4		Months	Along Columns				
5		Accounts	Along Rows				
6	OLAPTable						
7		Total Year	January	February	March	April	May
8	Margin %	72.30769	28	28	0	100	0
9	Net Sales	1625000	250000	375000	0	1000000	0
10	Cost of Sales	450000	180000	270000	0	0	0
11	Gross Profit	1175000	70000	105000	0	1000000	0

Now, to see the dynamic connection *back to* the PowerOLAP cube:

- Return to the PowerOLAP *Accounts by Months* slice (showing *Canada* as the Page Member).
- Press **F9** to update PowerOLAP.
  The Slice appears as follows:



The data you entered in the Excel worksheet is now reflected in the PowerOLAP database. Because PowerOLAP's function connecting to the worksheet (OLAPTable, in this case) is bi-directional, you can enter data in either Excel or PowerOLAP and select **F9** to update. (Note you can not write into Aggregate member spreadsheet cells or cells governed by a Cube formula, just as in a Slice).

The strength and power of the spreadsheet connection to PowerOLAP cubes are central to the use of the product: PowerOLAP “disburdens” Excel of its calculation tasks—hierarchies /aggregate weights /formulas are calculated in PowerOLAP’s engine, across specifiable multidimensional data ranges; further, PowerOLAP relieves users/organizations of the difficulties of maintaining hundreds or more linked spreadsheets.

Defining Cube References

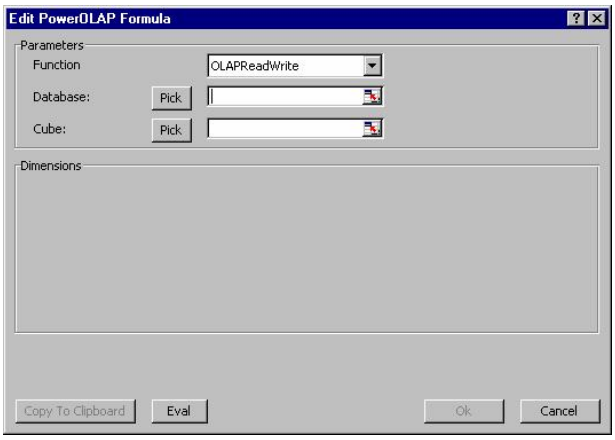
The bi-directional connection shown so far is OLAPTable function (shown in cell A6). In fact, when you click on cell **A6** in the worksheet, you will see in Excel's formula bar the single formula that references all the worksheet cells that connect to data in the PowerOLAP cube:

=OLAPTable(B1,B2,B5:R5,A6:A9,C3)

The OLAPTable function is one of many functions you can use to dynamically link data between a worksheet and a PowerOLAP database. (The other PowerOLAP functions for creating a worksheet from a Slice—OLAPReadWrite and OLAPPivot—and their differences, are discussed in the *PowerOLAP User Manual*.)

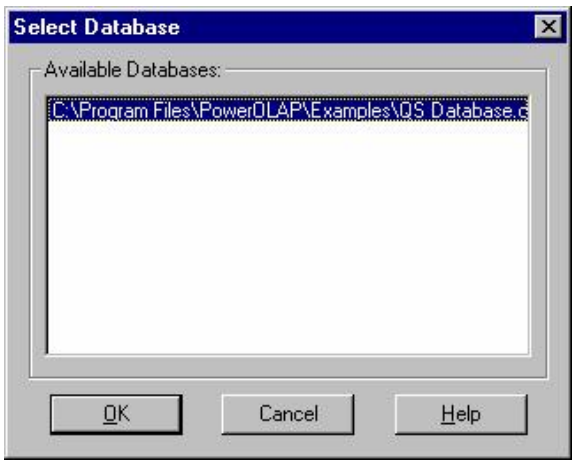
The next exercise will demonstrate how to define a Cube reference that returns a pertinent value into a cell by using the OLAPRead function.

- Select cell **D13** in Excel in the current worksheet. In this cell, you will define a Cube reference formula that shows the *Gross Profit* for *United States* in *February* (in order to make a quick visual comparison to *Gross Profit* for *Canada*, which appears in cell D9).
- From the Excel menu bar, select **PowerOLAP**, **Edit Formula**. The Edit PowerOLAP Formula dialog is displayed:



- Select **OLAPRead** from the top drop-down menu, to the right of *Function*.
- Press the **Pick** button,

, next to *Database*.
  The Select Database dialog box is displayed, as below:



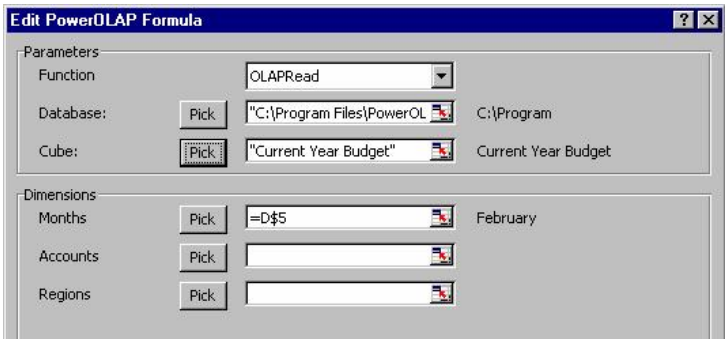
The current database—which contains the value you want—is selected.

➲ Click **OK**. You are returned to the Edit PowerOLAP Formula dialog.

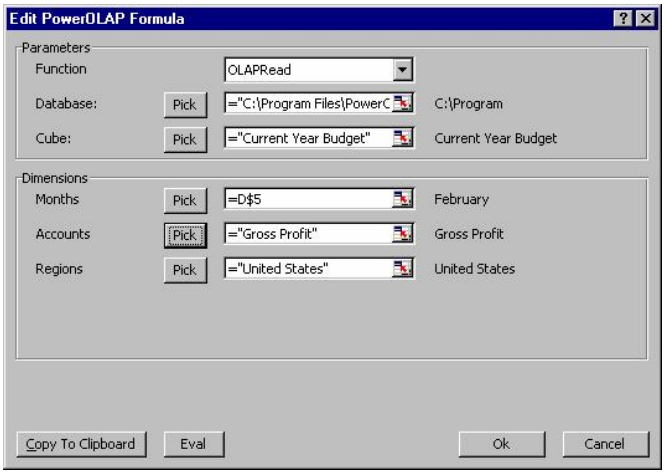
➲ Press the **Pick** button,

, to the right of *Cube*. The Select Cube dialog box is displayed. Likewise, this Cube contains the value you want to be reference into the Excel cell, D13.

➲ Click **OK**.
Again, you are returned to the Edit PowerOLAP Formula dialog box. Now you have defined several of the parameters of the Cube reference formula, as shown in this detail of the dialog box:



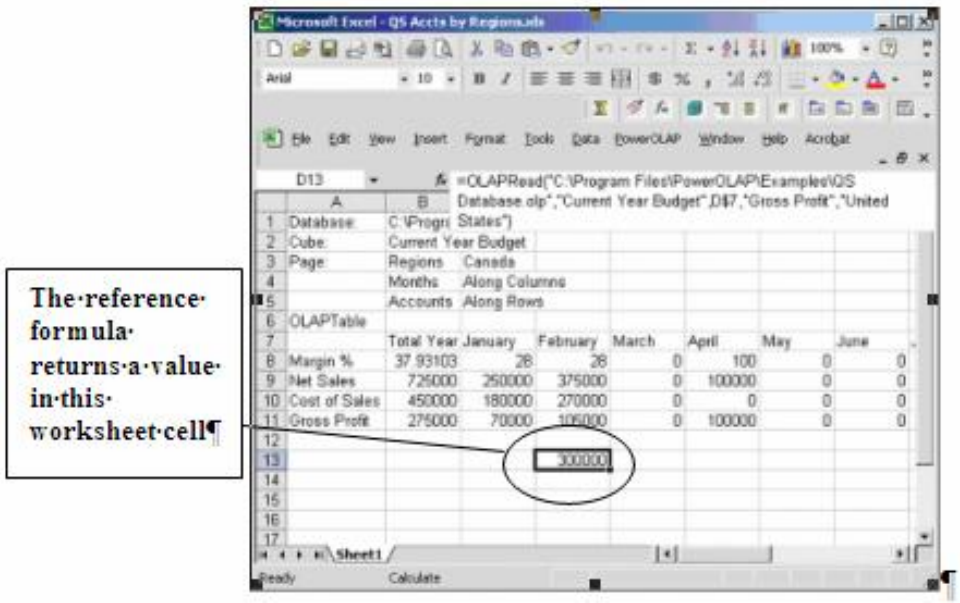
Note that the Dimensions area now displays text boxes for you to enter choices for the *Months*, *Accounts* and *Regions* dimensions. In fact, *February* has been "pre-selected" for you. If you wanted another *Months* member for your formula, you would press the **Pick** button to make a different selection. Since you do want to select *February* data, continue to the *Accounts* and *Regions* dimensions. Use the Pick button and the corresponding Select A Member dialog boxes to choose data for **Gross Profit** and **United States**, respectively.
After you have made these choices, the Edit PowerOLAP Formula dialog box will look as follows:



➲ Click **OK** to update Excel with the new formula reference.
Your dynamically connected Excel spreadsheet will appear as in the following figure:

You now have a ready view of the *February, Gross Profit* for *United States* within a dynamically connected spreadsheet that shows figures for *Canada*. Now, whenever *February*, *Gross Profit* for *United States* (or for that matter, *Canada*) changes, it will be reflected in this worksheet. You can save the current Database, with changes you have made, from within Excel:

➲ Select **PowerOLAP, Save Current Database in Modeler**.
The PowerOLAP database is saved but not closed.


The reference formula returns a value in this worksheet cell
Microsoft Excel - QS Accts by Regions.xls
File Edit View Insert Format Tools Data PowerOLAP Window Help Acrobat
D13
=OLAPRead("C:\Program Files\PowerOLAP\Examples\QS Database.olp","Current Year Budget",D$7,"Gross Profit","United States")
Database C:\Progr
Cube Current Year Budget
Page Regions Canada
Months Along Columns
Accounts Along Rows
OLAPTable
Total Year January February March April May June
Margin % 37.93103 28 28 0 100 0 0
Net Sales 725000 250000 375000 0 100000 0 0
Cost of Sales 450000 180000 270000 0 0 0 0
Gross Profit 275000 70000 105000 0 100000 0 0
300000
Sheet1
Ready
Calculate

You now have a ready view of the *February, Gross Profit* for *United States* within a dynamically connected spreadsheet that shows figures for *Canada*. Now, whenever *February*, *Gross Profit* for *United States* (or for that matter, *Canada*) changes, it will be reflected in this worksheet. You can save the current Database, with changes you have made, from within Excel:

➲ Select **PowerOLAP, Save Current Database in Modeler**.
The PowerOLAP database is saved but not closed.

Closing a Database

The Close Database command located on the File menu in PowerOLAP closes an open database. When you have completed work within one database, you may still wish to work with another database. You must first close the currently open database before opening another database.

To close an open database:

- Select **File, Close Database**.
  If any Slices are open, PowerOLAP will prompt you to save Slices.
- Clicking **Yes** will save all database changes to disk and close the database file.
  Clicking **No** will close the database file without saving any changes made to the database.
  In either case, all open Slices will be closed along with the database.

As noted earlier, you can save and close any dynamically connected worksheet as a normal XLS file. Upon opening such a worksheet, when you press F9, PowerOLAP launches, and a spreadsheet system with OLAP cubes behind it, is ready for online, optimized planning / analysis / reporting.

Summary of Quick Start Exercises

In the preceding pages you very quickly learned important basic concepts and fundamental functions of PowerOLAP, including:

- Creating a PowerOLAP database, the first step in building a Cube to model multidimensional data.
- Creating Dimensions, adding Members to those Dimensions, establishing a Hierarchy among Members (whether Detail or Aggregate), and assigning an Aggregate Weight to a Child member.
- Creating a Cube from Dimensions and their respective Members.
- Creating a Slice, arranging Slice dimensions, selecting Page members to view, and changing the layout of the grid within a Slice.
- Setting general and formatting preferences from the Edit, Options. Menu.
- Entering data in a Slice, and seeing how PowerOLAP automatically recalculates Aggregate members to reflect changes in value. Then, saving those changes to a database.
- Importing data from a file into a Cube.
- Creating Cube formulas.
- Creating a fully functional Excel worksheet from a Slice, and defining database reference formulas.
- Saving changes made from within Excel into the PowerOLAP modeler, and closing the PowerOLAP database, knowing that you can reopen it from a normal Excel worksheet.

Now that you have grasped the concepts and demonstrated these many functions, you are well prepared to use PowerOLAP in a production environment. For more detailed instruction on using PowerOLAP, and to learn additional features, see the *PowerOLAP User Manual*.

On the following pages, read more about two important advanced features of PowerOLAP: OLAP Exchange and PowerOLAP Services.

About OLAP Exchange

Note: This feature is enabled thro' Licensing. Check Edit/Options/Licensing to see if OLAP Exchange functionality is enabled on your system.

One of the key features of PowerOLAP is the back-end integration technology, OLAP Exchange, which bridges the gap between relational and multidimensional worlds. Through the rapid deployment of OLAP Exchange, organizations can propagate data dynamically between transactional and analytical databases—for advanced, online, real-time planning, budgeting, forecasting, analysis and reporting needs.

These next several pages provide an overview of the deployment of OLAP Exchange to create a Cube directly by sourcing data from, in the example shown, a SQL Server example database. (Please see the OLAP Exchange manual for a detailed explanation of the entire range of features and benefits of OLAP Exchange.) Keep these three important points in mind as you proceed:

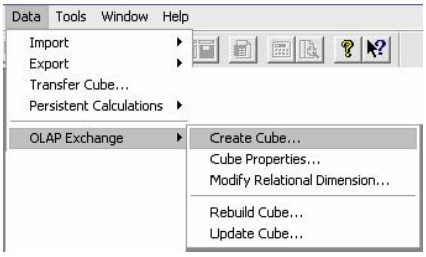
OLAP Exchange—apart from being able to source data from, for example, SQL Server tables—can reach other ODBC-compliant relational data sources. Thus, there is no "fixed" back end database or application that OLAP Exchange requires to create Cubes.

OLAP Exchange provides limitless customization options in the way it allows users to create Cubes. Put another way, OLAP Exchange does not force users to build so-called "standard" Cubes (for example, a reporting Cube from a General Ledger only). While, in theory, "standard" Cubes that can be built by "off the shelf" Cube-generating products sound like a good idea, in reality they can be extremely difficult to maintain and next to impossible to customize. OLAP Exchange overcomes this serious limitation by allowing rapid individual customization in the creation of the initial Cube—and in the creation of any number of subsequent Cubes required for planning, analysis and reporting.

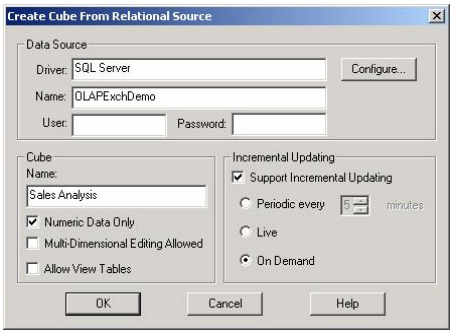
After creating Cubes via OLAP Exchange, you still enjoy all the features available in PowerOLAP for sophisticated business-specific modeling. As a simple example, assume you have created a six-dimension Cube from tables in an underlying accounting package; you can next create a "Forecast" dimension, and use the six already-created Dimensions to create a seven-dimension Forecast cube. Further, you can populate this Forecast Cube via Cube Formulas, and then create input Excel templates for key Forecast sales assumptions. You have already performed these steps in this manual—with Cubes created via OLAP Exchange, you can truly put to use all those capabilities to create a unique, powerful, sophisticated solution for your company!

The following is an overview of the steps to follow when creating a Cube via OLAP Exchange—again, these steps are explained fully in the OLAP Exchange manual.
To build a Cube from a relational source, you can begin by creating a new PowerOLAP database or you can create a Cube in an existing database; proceed, next, by selecting OLAP Exchange as follows:

➲ Select **Data, OLAP Exchange, Create Cube**.



The Create Cube From Relational Source dialog appears. In addition to naming the Cube and specifying security to access the Cube, in this dialog you will configure the OLAP Exchange Data Source (click on Configure) using Data Source naming conventions corresponding to the Driver to the source relational database. The following image shows a completed dialog, with SQL Server indicated as the data source.



Note some of the other selections in this dialog, including the bottom-right area concerning Incremental Updating. This feature enables OLAP Exchange to integrate only the incremental data that is recorded in an underlying relational database, rather than needing to rebuild a Cube for each update. Note that Incremental Updating can be done periodically, live or (as shown in the figure), on demand.

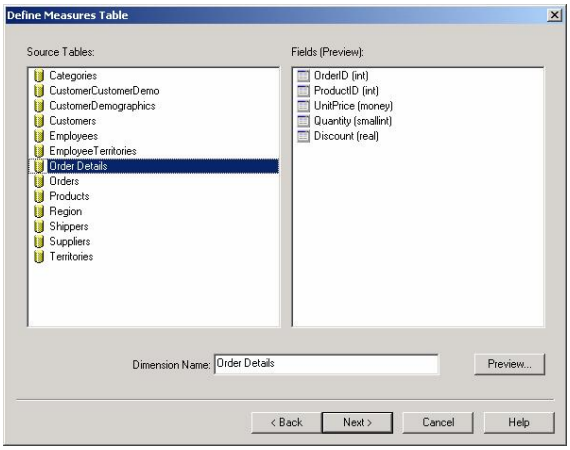
➲ Click **OK**. You will next see the Dimensions dialog—blank, since you have just started building the Cube.

➲ Click **Add** in the Dimensions dialog.

In the Select Dimension dialog, the Measures Dimension radio button will be selected—the Measures dimension is the first Dimension you will build. This Dimension will be based on the table in the underlying database containing, most typically, the Fact Data values that you want to analyze and report on, and which you may use as the basis for further modeling, e.g. for budgeting/forecasting.

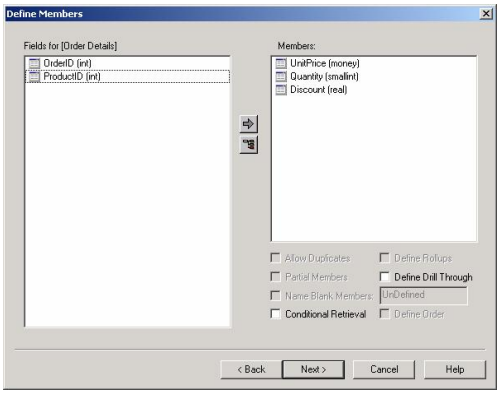
➲ Click **Next** in the Measures Dimension dialog.
The Select Measures Table dialog appears.

Here you see for the first time PowerOLAP reaching through from the multidimensional world (OLAP) to the relational word—all the underlying tables are shown in the left-hand pane.



By selecting table on the right, you will see the fields from that table in the right-hand pane—you can even click on the Preview button to see the first few rows of the table and its fields. This will enable you to determine if in contains the Fact Data in question.

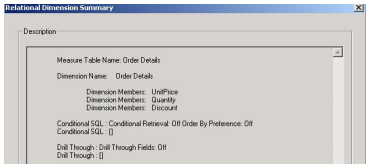
➲ Click **Next** after determining the Measures table.
The Define Members dialog appears.



The Define Members dialog is where you will make the selection of fields that supply the Fact Data points for the Cube—for example, in the figure above, Unit Price, Quantity, and Discount.

Note also the options at the bottom right: these enable you to create an "on the fly" Conditional Retrieval statement for bringing over the Fact Data (e.g., Unit Price > 1) and to indicate what fields you wish to see in a "drill-through" from an eventual Slice or even an Excel spreadsheet. (If you enable Drill Through, the next dialog will allow you to specify those fields.)

➲ Click Next.
A Relational Dimension Summary dialog appears (as in the detail figure below). This dialog sums up the "logic" of the Dimension you just created.


Relational Dimension Summary
Description
Measure Table Name: Order Details
Dimension Name: Order Details
Dimension Members: UnitPrice
Dimension Members: Quantity
Dimension Members: Discount
Conditional SQL : Conditional Retrieval: Off Order By Preference: Off
Conditional SQL : []
Drill Through : Drill Through Fields: Off
Drill Through : []

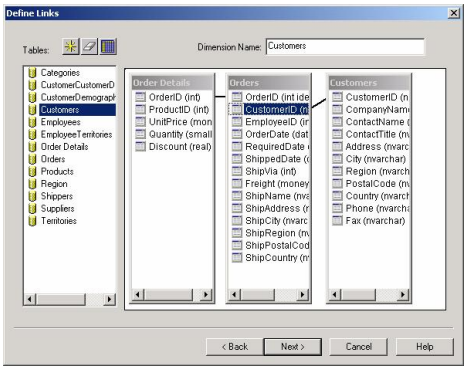
This completes the creation of the Measures dimension in this example. Since this is a Sales Analysis cube (built from the example Northwind SQL Server database). Next you will create Customer and Product dimensions—in order, from the Cube, to analyze and report on the Quantity, Unit Price and Discount of the products your customers purchased.

➲ Click **Finish** in the Relational Dimension Summary dialog.
The Dimensions dialog appears with the first—i.e., Measures—dimension listed.

➲ Click **Add** to proceed to the creation of the next Dimension.
The Select Dimension appears again. This time New Dimension is indicated (radio button enabled). Click **Next**.
[If you were working on a second or further Cube, you could select Shared Dimension, i.e., one that was created for use in a Cube created previously.]
The Define Links dialog appears on screen.

Let us proceed under the assumption that you next wish to create a Customers dimension. The objective—which will be accomplished in the Define Links dialog—is to "work back" from the eventual table that will supply the data for a Customers dimension to the Measures table. This is a good example of the need to link via other tables back to Measures: in this

case, knowing something about the logic of the underlying tables, you could reach Measures by creating links via the Order table.
In other words, first, with Order Details table already showing in the Define Links dialog (this crucial Measures table always shows as the first table in the link schema), you would double-click on the Orders table, to make it show in the dialog; then, do the same for the Customers table.

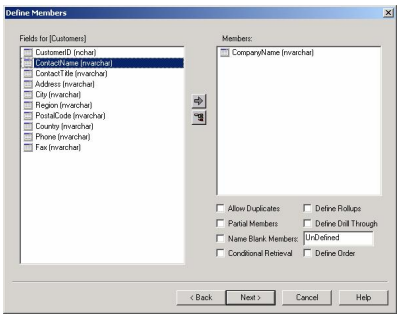


➲ Next, to define links between the tables, hold down the top left of your mouse and "draw links" between common fields—akin to creating an ad hoc "star schema".

In the preceding figure, you see that the Order ID field in the Order Details (i.e., Measures) table has been linked to the Order ID field in the Orders table; and the Customer ID field in the Orders table has been linked to the Customer ID field in the Customers table. In this manner, you can now proceed to use the last table (Customers) to supply the "logic" for the Customers dimension.

➲ After the proper links have been created, click **Next**.
The Define Members dialog appears.

In this dialog (like the Define Members dialog for Measures), you will want to specify the field that will supply Members for the Customers dimension. A logical choice is the CompanyName field—thus, this field has been moved to the right-hand pane in the following figure.



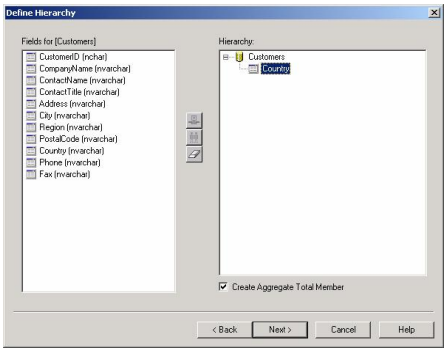
Note in the dialog the further options available at the bottom right of the dialog. These options are explained in detail in the OLAP Exchange manual. It is worth noting here, though, the customization options that OLAP Exchange affords for dimension-creation: for example, the ability to create your own rollups (consolidated Members that you name); or, creating Member names in a certain Order or composed of a fixed number of characters.

➲ After selecting the field to supply Member names, click **Next**. The Define Aliases dialog appears. In this dialog you can select a field from which to create coordinating unique

Alias names for the Member Names—very useful if you wish to report on, for example, Customer ID numbers along with Customer names

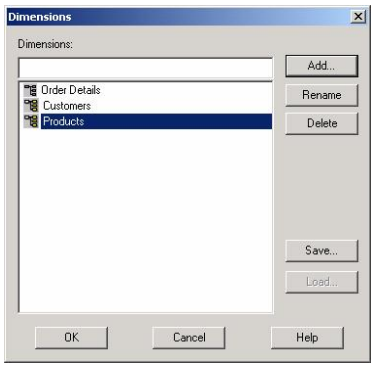
➲ Click **Next**. The Define Properties dialog appears. Here, too, you can "leverage" the logic of underlying tables, to report on Properties (which need not be unique) characteristics of Members.

➲ Click **Next**. The Create Hierarchy dialog appears.
In this dialog, also, you can construct the Dimension in such a way to ultimately analyze/report on data according to the hierarchical logic of the elements in a table—for example, as shown below, your Customers may exist within Countries (or Regions of Cities), which is recorded in the underlying table. In the Define Hierarchy dialog, you would drag Country under Customer in the right-hand pane to create this hierarchy in the PowerOLAP dimension.
Note too that there is a provision (checkbox, bottom right) to create an Aggregate Member for the Dimension—here, it will be Total Customers.



➲ Click Next. The Relational Dimension Summary for the newly created Dimension appears.

When you click **Add** again in the Dimensions you start the process of creating another Dimension via OLAP Exchange. Assume, for example, that you wish to create a *Products* dimension. You will go through the same steps, which are:
Define Links
Define Members
Conditional Retrieval (if checked)
Order Preference (if checked)
Partial Fields (if checked)
Drill-Through Values (if checked)
Define Aliases
Define Properties
Define Hierarchy
Define Rollups (if checked)
Relational Summary
In this example, assume that you have completed the creation of a *Products* dimension. After seeing the Relational Summary, you would click **Finish**, and be returned to the Dimensions .dialog:



At this point, if it is logical that a Cube with 3 dimensions will yield worthwhile results for your planning, analysis and reporting requirements (and for this example, it certainly will!), you can click OK to create the Cube. [Note that, upon clicking OK, an option appears to "save the template" of the Dimensions you created—this will be useful for additional Cubes you might create with those Dimensions.]

The progress bar will appear onscreen until the Cube creation process is complete. When finished, a message box appears indicating the number of data points in the Cube.

Having created the Cube has, you will successfully have "bridged the gap" from relational to multidimensional worlds—and, even better, you will have all the modeling capabilities described in the initial exercises in this manual. Assuming you create spreadsheet reports or templates from Slice of the Cube, you will enjoy the capability of an online, dynamic data flow from underlying relational database tables through to your desktop. With a few clicks, you can Update Cube for Incremental Data that has been recorded in relational tables (if the Cube supports Incremental Updating), or Rebuild the Cube completely. These and other, more advanced functions are accessible via Data, OLAP Exchange on the PowerOLAP application window

Note that OLAP Exchange also provides for sending OLAP data back to relational tables—in this way, OLAP Exchange is truly bi-directional. Please consult the OLAP Exchange manual for full details.

About PowerOLAP Services

Note: This feature is enabled thro' Licensing. Check Edit/Options/Licensing to see if Analysis Services functionality is enabled on your system.
:
PowerOLAP Version 7 includes a dynamic front end capability to Microsoft Analysis Services cubes and to other OLE DB for OLAP – compliant cubes. This capability is known as PowerOLAP Services.
PowerOLAP Services can be employed for a number of critical uses, particularly given the power afforded the user in the PowerOLAP Cube Modeler. Dimensions can be added "going forward" with a few clicks, just as you have demonstrated in the preceding exercises. Data can be sent into new Cubes created in PowerOLAP, via the Formula writer (which you also demonstrated previously) or via the Transfer Cube feature (for this feature, see the PowerOLAP User Manual).
Furthermore, with the Cubes now in PowerOLAP, you can use all familiar spreadsheet capabilities, with no loss of the everyday and powerful features Excel affords you.
A full explanation of working with "Linked Cubes" from Analysis Services occurs in the PowerOLAP User Manual. What follows is a brief description of the procedure, beginning with the configuration a connection from PowerOLAP to the Microsoft Analysis Services environment.
Assuming you have created a new PowerOLAP database, or wish to create a Linked Cube in an existing database,

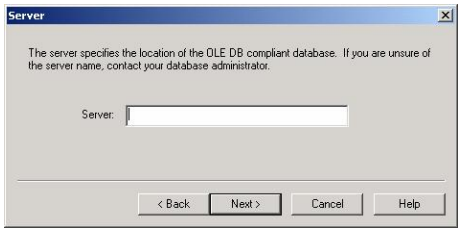
➲ Select **Tools, PowerOLAP Services, New Linked Cube**

You will next be prompted for the Cube Provider:

Provider
To link to an OLE DB compliant server, you must first select a valid Provider. The provider allows PowerOLAP to communicate to a specified database.
Provider:
MSOLAP
Description:
Microsoft OLE DB Provider for OLAP Services
< Back
Next >
Cancel
Help

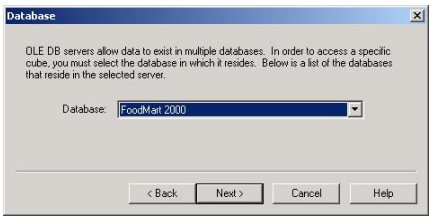
➲ Select a Provider (e.g., MSOLAP, as above), and click Next.

The next dialog concerns the Server name:


Server
The server specifies the location of the OLE DB compliant database. If you are unsure of the server name, contact your database administrator.
Server:
< Back
Next >
Cancel
Help

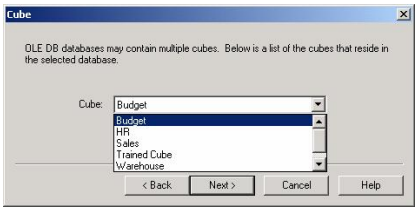
➲ Copy/paste or type in the **Server name**, then click **Next**.

PowerOLAP finds and connects to the Server; available databases on the Server are shown in the ensuing dialog.

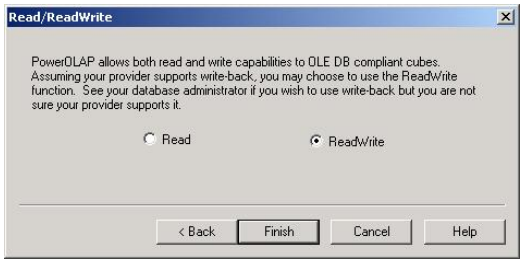

Database
OLE DB servers allow data to exist in multiple databases. In order to access a specific cube, you must select the database in which it resides. Below is a list of the databases that reside in the selected server.
Database:
FoodMart 2000
< Back
Next >
Cancel
Help

➲ Select a **Database** (as above, FoodMart 2000), then click **Next**.

The Cubes available on the database are shown in the ensuing Cube dialog (drop-down shown below). You access one Cube at a time to become a PowerOLAP Linked Cube.



➲ Click **Next**. The Read/ReadWrite dialog appears:


Read/ReadWrite
PowerOLAP allows both read and write capabilities to OLE DB compliant cubes. Assuming your provider supports write-back, you may choose to use the ReadWrite function. See your database administrator if you wish to use write-back but you are not sure your provider supports it.
Read
ReadWrite
< Back
Finish
Cancel
Help

The final step is to select whether the method of creating the Linked Cube should be a Read or a ReadWrite formula:
OLEDBRead will read data from the data source, but the data is not editable—thus it is "Read-Only".
OLEDBReadWrite also brings in data by reading the data source, but it also makes the data editable, so that it is possible to write back to the source if the cell is a Detail, i.e., leaf level cell.

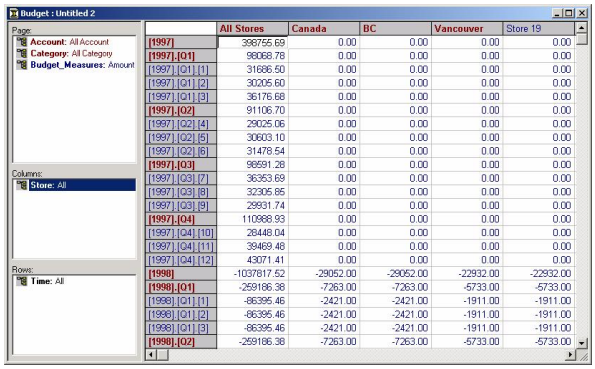
➲ Thus, select **Read** or **ReadWrite**; next, click **Finish** to complete the Linked Cube creation.

The progress bar will appear onscreen until the process is complete. When finished, the following message box appears.

➲ Click **OK**.
You are returned to the PowerOLAP main application window

You can next select Model, Cube to see the Linked Cube itself, or select Model, Dimensions to see what Dimensions have been created.

➲ To determine what a Slice looks like, select **Slice, New**:



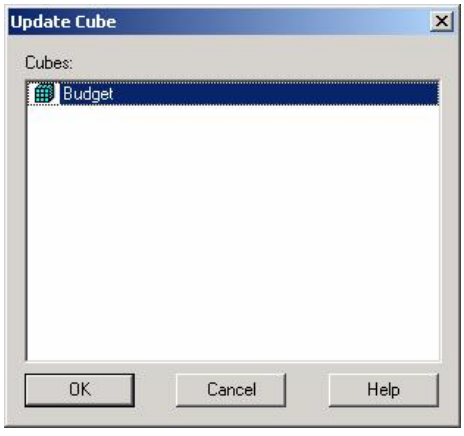
All the Fact Data in an MS Analysis Services cube has also been delivered to the PowerOLAP Cube, and appears in the Slice.

To update the Fact Data in Cube, all you need to do is hit F9 (whether you are in PowerOLAP or an Excel worksheet created from a Slice. Concerning "write back": provided the Cube was created using the ReadWrite capability and the cell into which you type data is a all Detail (i.e., leaf) Members and the Cube is enabled for "write-back" (and you have the requisite rights to do so), you can hit F9 also to send that data point to the MS Analysis Services cube.

Finally, the Update Linked Cube function allows you to update the PowerOLAP cube with any new Meta Data and Fact Data changes that were made to the "provider" Cube (e.g., an MS Analysis Services cube):

➲ Select Tools**, PowerOLAP Services, Update Linked Cube**.

The Update Cube dialog appears:



The dialog lists the Cubes that were created as a linked cube.

➲ Select the Cube(s) to rebuild and click OK.

Note that this action deletes all existing Members from the current Cube and rebuilds them from the OLE DB for OLAP – compliant source cube, including all Meta Data and associated Fact Data.

This completes the overview of PowerOLAP Services, the feature that shows how PowerOLAP—along with being an optimized cube-builder itself—can function as an especially strong front end for MS Analysis Services and other OLE DB – for OLAP systems as well.